Computer Science and Information Systems 14(2):393–421 DOI: 10.2298/CSIS160229007W

Effort Estimation in Global Software Development - A Systematic Review

Dilani Wickramaarachchi[1] and Richard Lai[2]

[1] Department of Industrial Management, Faculty of Science,
University of Kelaniya, Sri Lanka
dilani@kln.ac.lk
[2] Department of Computer Science and Information Technology,
La Trobe University, Melbourne, Australia.
r.lai@latrobe.edu.au

Abstract. Global Software Development (GSD) is becoming increasingly prevalent, with software development teams being distributed around the world and working in collaboration with partner companies despite geographic and time differences. The main advantage of GSD which makes it attractive is the greater availability of human resources at lower costs. However, there are several disadvantages which are caused by the distance which separates the development teams. Coordination and communication become more difficult when the software development teams are located in different places, resulting in hidden costs involved in this process. As such, the effort estimation models used for collocated software development are inadequate for estimation in GSD. Thus, effort estimation in GSD is becoming an important area of research. Many researchers have focused on effort estimation in GSD over the last decade. This paper presents the findings of a systematic review of the related literature by summarizing the hidden costs in GSD, and discussing the open research issues in effort estimation in GSD.

Keywords: Global Software Development, Business Process Outsourcing, Effort Estimation, Global Software Engineering, Distributed Software Development, Offshoring, Insourcing, Onshoring.

1. Introduction

Global software development can be defined as any aspect of software engineering that involves the combined efforts of software professionals in different locations (separated by significant distances [1]/ distributed beyond the limits of a nation[2]). Significant distance can be further described as dispersed across national boundaries. Sakthivel calls it offshore system development: system development in another country [3]. Global software development (GSD) is also known as Globally Distributed Software Development (GDSD), Global Software Engineering (GSE) [4] collaborative software development [5] and global virtual teams development[6].

With the development of globalisation, the significant influence on software development has been recognised by many studies today. In this era, Business Process Outsourcing (BPO) has become a natural evolution of the global market [7]. Tight budgets and a shortage of resources and time has motivated many enterprises to start looking for partners outside. With the help of the Internet, it is now possible to share information with anyone around the world. This opens the way to locate digital services around the world. However, many people are not aware that most of the software we use today is the result of intense cooperation between project managers, architects, analysts, developers, testers and other people who are located all around the world [8]. Currently, it is becoming necessary for companies to offshore Information Technology (IT) services in order to survive in a competing market. This argument is supported by the survey conducted in U.S. in 2009 which shows the value of the offshore software development market has increased 25-fold over the last 10 years [9].

However, GSD brings both opportunities and challenges [4, 10, 11]. The main factors which motivate organizations to utilize GSD are (i) low-cost labour, (ii) access to international talents, (iii) reduction of product cycle time, and (iv) opportunity for innovation [9, 10]. Conversely, the major challenges of GSD are the different time zones, language, and cultural differences which makes communication and coordination much harder [10, 12]. One of the main problems in communication is negotiation with globally distributed teams which increases project planning and tracking costs [12]. A major challenge in coordination is the synchronization of virtual teams [12]. Further, globally distributed teams have fewer opportunities for informal communications that are essential to coordinate development effort, manage task dependencies, and maintain awareness [12].

In spite of the growing attention on GSD, there is a limited understanding of how to ensure GSD is a success and why it fails [13]. According to Aron and Singh [14] several studies illustrate that half of the organizations which shift processes offshore fail to generate the expected financial benefits and some of them are failing completely [15]. GSD projects and case studies range from announcements of remarkable victory to total collapse [4]. The cost-benefit trade-off in GSD is still not fully understood [16]. No research so far has imparted a clear vision of the true amount of investment necessary to make global software projects work [4]. Hence, offshoring not only offers opportunities to gain extra profit, but also brings extra threats that adversely influence profits. Despite these threats, offshoring is becoming a norm in the software industry [17]. Therefore, the challenge is how the threats can be reduced, in order to increase the chance of high profits. Rephrasing this question provides an overview of our topic of research: utilizing effort estimation in GSD to identify the potential benefits and threats in advance.

Due to the increasing popularity of GSD, effort estimation in GSD has become an important area of research. This paper presents the findings of a systematic review of the related literature by summarizing the hidden costs in GSD, and discussing the open research issues in effort estimation in GSD.

1.1. Global Software Development Research

The need for research in GSD is growing with an increase in the number of internationally distributed software organizations. Carmel and Agarwal [18] report that practitioners are experimenting and quickly adjusting their tactical approaches to leverage global software development risks, because the methods used in collocated projects have limited applicability in the GSD environment. Although some theories and practices have been researched and developed, the art and science of global software development is still evolving [17]. This means the concept of GSD is still unclear.

Terminology/ Taxonomy used in GSD: Wider acceptance of GSD by the software industry has resulted in different classification schemes depending on the way it has been absorbed by the industry. Global software work is enabled through various collaborative forms such as inter-organizational outsourcing, intra-organizational offshoring, intranational near-shoring, to name a few. However, since GSD scenarios are diverse, what works in one context might not be directly applicable in another [19]. The classification and terminology used in GSD is not yet properly standardised. Thus, it would be helpful for future researchers to be familiar with classification schemes and terminology to have a better understanding of the research problems. To improve the readability of the paper, the classifications and terminologies used in this paper are given below.

Simplified Matrix of GDSD Business Model: The GDSD business model matrix represents the relationship structure and geographic location of the work. [20].

Control/ Ownership/ Relationship structure	Onshore/National	Offshore/ International
	Onshore Outsourcing OR Outsourcing	Offshore Outsourcing
	Internal Domestic supply	Captive/Internal Offshoring OR Offshore Insourcing
	Geographic Location	

Fig. 1. Simplified Matrix of the GDSD Business Model

As our main focus is GSD, the two models in the second column in the above matrix will be the area of interest in our study. In offshoring, there are different types of cooperation. Fabriek [8] introduced three different cooperation types and their interactions are illustrated in Fig.2.

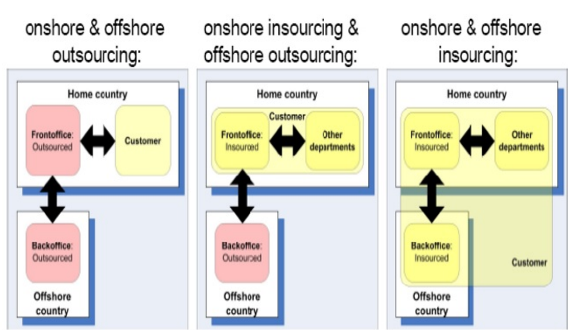

Fig. 2. Overview of different cooperation types[8]

Even though it may not be very common, another cooperation type can be added, this being onshore outsourcing and offshore insourcing.

As stated by Herbsleb and Moitra [2], the processes employed in offshore outsourcing might be different to those employed in internal offshoring, and the characterization in this case could make a difference for the practice of GDSD. Moreover, research conducted in one type of distribution is not necessarily valid for all types of GDSD. Therefore, the different characteristics of GSD that will be encountered in this study are explained in Table 1.

2. Systematic Review Procedure

This systematic review has been carried out following the structured procedure proposed by Kitchenham [25, 26]. It permits the identification, evaluation and interpretation of all the available relevant studies related to particular research questions. Thus, it provides results of a high scientific value by synthesizing existing work according to a predefined strategy.

2.1. Question Formulation

Questions have been formulated to communicate the message in logical way, emphasizing the need for effort estimation in GSD, the difficulty of effort estimation in an offshore environment, the available models and the significance of the research problem.

The research questions that guide the systematic review are:

1. What initiatives have been carried out in relation to effort estimation in GSD?
2. What are the additional challenges involved in the process of effort estimation in GSD?
3. What are the effort estimation models for GSD available in the literature?
4. What systematic studies have been conducted on effort estimation in GSD?

The ultimate goal of this systematic review is to provide a critical analysis of recent research by identifying the importance of effort estimation, the estimation models for GSD in the literature and future directions for improvement.

2.2. Source Selection

In order to carry out a comprehensive analysis, the search strings were established by combining the key words through logical connectors AND and OR.

The studies were obtained from the following search sources: IEEE Xplore, ACM Digital Library, Science Direct, Springer Link, Wiley Interscience and Google Scholar. The quality of these sources guarantees the quality of the study. Further, some white papers on hidden costs in GSD are referred to because of the unavailability of this information from the above mentioned sources.

In addition, as a technique to identify the most relevant articles whose topic does not match with the key words of the search criteria, cited articles of the most relevant papers were searched and filtered. Through this approach, relevant articles which were missed by the search strategies were highlighted.

Table 1. Different characteristics of GSD

Characteristic	Categories	Description
Ownership/ Collaboration mode	Inter-organization/ Offshore outsourcing	Software development is moved to an external third party in another country [5]
	Intra-organization/ Internal offshoring, offshore insourcing	Software development is moved to a division of a specific company established in another country [5]
	Offshoring	A generic term when the relationship of the overseas company is unknown [5]. (Transfer of organization function to another country [21])
Number of distributed sites	Two, three, four, etc.	The complexity of communication. coordination and control in GSE projects is dependent on the number of distributed sites. Increasing the number of collaboration partners increases the number of sources of threat and also the complexity of trust achievement [22]
Location	Geographic Distance: Near shore Far shore	Near shoring: jobs are transferred to geographically closer countries [21]. Scaling these distances from near shore to far shore makes a difference to how the teams interact. The impact of two collaborated remote teams is symmetric [19]
	Time Zone: Overlapping hours in collaborating sites	The impact of two collaborating remote teams is asymmetric
Perspective	Customer/client Supplier/ provider Collaboration/relationship	The perspective of reported empirical findings may vary between suppliers, originators or collaboration in general [23]
Reason for GSD	Cost, resources, speed, new market	Studies show that there is a dependency between the reasons for implementing GSE scenarios and the actual success of collaboration [7]
Development Process	Extended work bench model	Centralized project management and control [24]
	Systems of Systems	Hybrid centralized/distributed management [24]

2.3. Studies Selection

The inclusion criteria for determining whether a study should be considered relevant were based on the title, abstract, keywords and the conclusion. In some cases, it was necessary to read the entire document to determine its relevance.

3. Results

3.1. Q1. What initiatives have been carried out in relation to effort estimation in GSD?

Significant research on software effort estimation began in the early 1960s and since then, it has been evolving [27]. As software grew in size and importance, it also grew in complexity which makes effort prediction much harder [27]. Accurately predicting software development effort is a crucial concern of many organizations today [28]. Underestimating development cost and schedules can lead to a massive loss and overestimating can result in a waste of resources and missing opportunities.

The main drivers of GSD are globalization and outsourcing [1], and in order to survive and succeed in todays competitive software industry. GSD has become a necessity [29]. GSD can be considered as a part of the trend of globalization of business operations [29]. Most software development companies are moving from traditional centralized local development to distributed global development [15]. This evolution is taking place due to the availability of high quality and low cost software professionals in different countries. It further encourages communication between dispersed groups [15] by increasing knowledge of each others work while accelerating the time to market [4].

Although, for several decades, there has been a growing trend to utilize GSD, it is hard to find any published documented evidence that proves GSD reduces the total software development effort and cycle time [12]. Researchers are divided on the impact of software offshoring, and contradictory arguments have been espoused. For example, some claim that the follow-the-sun model helps to reduce cycle time [30] whereas others claim it is a quick-and-dirty strategy that converts a scheduling problem into a quality disaster [12]. Therefore, the true benefits of GSD must be properly understood if we wish to fully realize them [10].The cost-benefit trade-offs of GSD are still not well understood [31], but it is very important to clarify this in order to estimate the effort required in GSD.

Benefits worth the risk in offshore software development [32] have become a widespread topic in the discipline. A survey and analysis of the literature on information systems outsourcing in [33] assesses the research done up to 2004 in IS outsourcing.

It is important to identify the true impact (positive and negative) of software offshoring to estimate the real effort necessary for software offshoring. Different opinions given by scholars as reasons to offshore are summarized in Table 2 (Positive impact + , Negative impact -).

An analysis of the literature shows that the main reason for software offshoring is the cost reduction by paying lower wages, but we could not find any proper method to assess whether the total cost of GSD is less than the collocated software development. According to Table 2, most researchers generally agree that cost reduction, having access to a large labour pool and a wider market are opportunities in software offshoring whereas, contradictory arguments have been given by different researchers regarding software quality and

Table 2. An overview of reasons to offshore

Literature	**Salary saving**	**Large skilled labor pool**	**Closer proximity to market and customer**	**New opportunities**	**Reduced cycle time**	**Innovation & best practice**	**Quality improvement**
A. Almeida, et al. (2009)[34]					- (increased)		
A. Khan and Z Muhammad (2009) [35]	partially realized	partially realized	partially realized			partially realized	
C. Ebert and P. De Neve (2001) [11]	+				+	+	
Chan, K. and L. Chung (2002) [36]							+
D. Damian and D. Moitra (2006) [17]	+	+	+	+			
D. Damian et al. (2003) [37]	+	+					
D. Farrell (2006) [38]		+					
D. Smite (2007) [7]					+		
D. mite and J. Borzovs (2008) [4]	+	+					
Davis, G. et al. (2006) [39]	+						
E. Carmel and R. Agarwal (2001) [18]	+	+					
E. Conchir et al. (2009) [9]				+			
E. Conchuir et al. (2006) [10]	partially realized	partially realized	partially realized		not realized	partially realized	
G. Seshagiri (2006) [12]		+					
H. Holmstrom, et al. (2006) [40]	+	+	+				
J. D. Herbsleb and R. E. Grinter (1999) [30]					+		
J. Herbsleb et al. (2000) [41]		+	+		+		
M. Bass and D. Paulish (2004) [42]				+			
M. Fabriek et al. (2007) [8]	+					+	
M. Maznevski and K. Chudoba (2000) [6]					+		
N. Huda et al. (2009) [15]	+				+		
R. Aron and J. Singh (2005) [14]		+	+		+		
R. Grinter et al. (1999) [43]		+	+				
R. Ravichandran and N. Ahmed (1993) [44]	+	+		+			+
S. Krishna et al. (2004) [45]	+	+					+
S. Sakthivel (2005) [3]	+	+					
W. Aspray et al. (2006) [46]	+						
W. DeLone et al. (2005) [13]		+	+				
J. Herbsleb et al. (2001) [47]	-				-		

the cycle time of offshore development which incurs some cost. Despite this, it is obvious that the primary motivation behind offshore development is cost [48].

Typically, only the potential benefits of GSD are discussed in many research articles and their realization is taken for granted [33]. An overview of GSD project case studies shows that the observed results can vary from tremendous successes to total failures [38]. However, GSD also introduces a number of challenges in relation to communication, coordination and control of the development process. These challenges arise due to the distances involved in three major dimensions of geographical, temporal, and socio-cultural issues. As a consequence, much research and practice has focused on trying to find ways to overcome these GSD challenges [10].

Table 3. An overview of GSD overheads

Literature	**Temporal distance results in misunderstand-ing/communication difficulties**	**Geographical distance results in informal communication/ communication difficulties**	**Different Socio-Cultural backgrounds results in misunderstanding**
D. E. Damian and D. Zowghi (2002) [52]	X		
D. mite et al. (2008)[19]	X	X	X
E. Conchuir et al. 2006 [10]	X	X	X
G. Seshagiri (2006)[12]			X
H. Holmstrom et al. (2006)[40]	X	X	X
J. D. Herbsleb and R. E. Grinter (1999) [30]		X	
J. Herbsleb et al. (2005) [53]	X		
J. Herbsleb et al.(2000) [41]		X	
M. Maznevski and K. Chudoba (2000) [6]		-	
M. Robinson and R. Kalakota (2004) [20]	X		
N. Huda et al. (2009)[15]			X
P. gerfalk et al (2008) [16]	-		
P. gerfalk et al., (2005) [54]	X	X	X
M. Rao (2004) [55]	X		X

According to [49], the key factors affecting the success of small and medium offshore software development companies are people, technical infrastructure, client interface, business infrastructure, and regulatory interface. Somewhat surprisingly, cost is not the most critical success factor. According to Cramton (2001), developing software in an offshore environment leads to problems such as poor decision quality, poor productivity and poor relationships [50]. Generic offshore characteristics can be referred to as distance measures because each of them implies a certain type of distance between the client and supplier [51]. Different views on GSD overheads found in the literature are presented in Table 3. A small time difference has a significant effect on the productivity of software development [56].

Table 2 and 3 clearly indicate that offshoring not only offers opportunities to gain extra profit, but also brings particular threats which can adversely influence profit. This shows that GSD has its own driving forces, restrictions, payoffs, and challenges [29]. Despite these threats, offshoring is becoming a norm in the software industry [17] which motivates us to study effort estimation in GSD. Further, the unavailability of a proper estimation model for GSD and the unsuitability of collocated software development effort models for GSD [57] was a further motivation for us to conduct the survey.

3.2. Q2. What additional challenges are involved in the process of effort estimation in GSD?

Organisations are eagerly embracing the concept of globally distributed software development to reduce cost, and increase efficiency and productivity which drive the business forward. Most measure the advantages of GSD by calculating the salary variation in different parts of the world and distribute the work to lower wage countries. But at the end of the project, it is sometimes true that the calculated cost benefits have not been realized. Thus, the reality of the cost deviation is not only related to the wage differences in collocated and distributed projects.

GSD is technically and managerially complex and presents a variety of challenges to be managed [40]. These challenges introduce some additional costs to the process of GSD. Additional costs are referred to as hidden costs, invisible costs, indirect costs, distribution costs and transition costs by different articles. In this paper, we use the term hidden cost to refer to additional costs. Hidden costs include travel time and expense for management oversight, review meetings, training, communication and coordination [12] etc. This cost can negatively impact productivity. In [12], Seshagiri claims that lower per-hour wages are offset by much lower overall productivity. Despite the significant advantage in labour cost, poor software quality drives up the cost [12].

Some firms have high expectations without gauging the actual effort involved [58]. However, it is becoming evident from experience that the reality of the situation does not simply lie in the salary difference. Several opinions of industry professionals on software offshoring are as follows: ”On paper, it looks extremely attractive. A Russian programmer charges 80 percent less than an American. But when you parse it all out, the total cost of offshoring a given IT job is generally comparable to getting the work done domestically, says Tom Weakland, a partner at management consultancy Diamond Cluster. It’s just that few companies are aware of these real costs. Most companies can’t accurately measure their productivity and costs prior to and after outsourcing,” says Weakland [59].

It is claimed that sometimes the quality of the software is not good in the case of offshoring. Empowered Software has discovered, programs developed by offshore outsourcers often contain more bugs than software programs domestically developed (usually 35 percent to 40 percent more). Further, Tom Weakland says ”If a company makes software for flying airplanes, I wouldn’t want it to be created with the priority of the deadline coming first and quality coming second,” [26].

It is clear that lower wages, higher quality and higher productivity are the benefits of GSD that should result in significant cost savings and improve the companys bottom line. Unfortunately, GSD projects do not always produce these positive objectives as the outcome, mainly due to hidden costs.

Table 4. Hidden costs involved in the process of GSD

Main cause	Brief Description	Hidden cost introduced
Physical distance	Physical distance is symmetric and it creates communication difficulties among remote teams.	Travel time, Review meetings,Training, Paying more employee for the same task
Time zone	Impact of time zone differences is asymmetric and variations in time zones changes the number of overlapping hours which limit the possibility of communication among remote teams.	Idle time of employees who cannot proceed while waiting for a reply from a remote team.
Language barrier	It creates misunderstandings among remote teams. This is a result of poor communication.	Poor quality, Rework, Bad image of the company
Management oversight [60]	Remote teams need to be managed and coordinated properly.	People (new jobs introduced), Process, Technology
Coordination	Compensate for the negative impact and increase the final performance.	Naming conventions, Standards, Planning, Inter-site coordination
Collaboration	Collaboration among remote groups involves some cost and it depends on the collaboration maturity.	Level of collaboration maturity
Cultural differences [60]	Variation in values in different cultures results in an extra burden on management.	Idle time in one site because of a public holiday in another country
Infrastructure [61]	The cost of the physical infrastructure and other support is usually an additional redundant cost. In most cases, the offshore outsourcing initiative does not lead to a substantial reduction in the physical infrastructure (buildings, power, etc.) at the enterprise.	In the case of offshore (captive, wholly-owned) BPO or call centre operations, these costs can be substantial. These costs are very location-specific and vary between locations within countries too.
Exchange Rate Fluctuations [61]	Currency fluctuation	
Vendor selection [60]	These selection costs include documenting requirements, evaluating the responses, and negotiating a contract.	Travel cost, Possibly a person working on it full-time, Legal costs, Time consuming process (time is money)

Hidden costs result in GSD not being as good as it appears to be. In addition to the known advantages of GSD, there are also some known and unknown serious problems. Most of the key problems/issues have been identified by researchers. Although these additional challenges in GSD have been discussed by many people in different circumstances, there is no proper mechanism by which to evaluate whether the benefits outweigh the challenges. So, it is necessary to identify the hidden costs of these challenges to determine whether they are measurable when estimating the cost involved in GSD. Table 4 lists the most significant hidden costs identified in the literature survey with the main cause and a brief description.

All these measurable and immeasurable costs should be considered when evaluating the total cost of offshoring. Measuring immeasurable costs is one of the biggest challenges in software effort estimation. Most of the software development cost is related to salaries and recent evidence shows that team-related factors can affect the projects performance and its cost [62]. Selecting an appropriate team size is a challenging task [63]. Larger teams facilitate a better distribution of skills, but lead to higher communication and coordination costs [64].

Q2.1. Are we really saving money or just shifting the cost burden? Because of these additional costs introduced in the process of offshoring, if the company adds up the costs at the end of the project, it might discover that the projected savings are not there. But, it will be too late to take action to correct it. So, there should be a method which can be used to estimate the total cost of the GSD before starting the project.

Table 5. Summary of the findings relating to offshore software costs

Summary of Findings Relating to Cost	Source
Only 54 percent of offshore agreements realized the expected cost savings.	Hirschheim and Lacity, 2000 [65]
One in every three offshoring contracts targeting cost reductions failed to meet expectations.	Caldwell, 2002a, 2002b
38 percent of the participants paid hidden costs, 31 percent stated that vendors became complacent once contracts were in place, 20 percent experienced greater than expected employee turnover, and 44 percent found that vendors did not have the capability to provide the expected level of quality and cost savings.	Deloitte, Touche, and Tohmatsu, 2003
Only 50 percent of offshoring in the near future will be successful.	Deloitte, Touche, and Tohmatsu, 2003
A study of 116 offshoring decisions found that 38 percent of these arrangements were successful, 35 percent were failures, and 27 percent had mixed results relative to cost, quality, flexibility and other considerations.	Kern and Willcocks, 2003
Half the organizations that shifted processes offshore failed to generate the financial benefits they expected.	Aron and Singh, 2005 [14]
70 percent of companies have had significant negative experiences and are offshoring with increasing caution and in a conservative manner.	Murray and Crandall ,2006[66]
In a survey of 50 companies, about 14 percent of outsourcing operations were deemed a failure.	Barthelemy, J.,(2001)[67]
Dun and Bradstreet Survey finds 50 percent of outsourcing relationships worldwide fail within five years, the principal cause being poor planning for new and evolving business processes	Bradstreet, D.(2000) [68]

Overall, this means the IT budget may reduce at the beginning but it does not ensure that total cost of the project will be low, as work quality, dissatisfaction and time delay may add more cost to the project. The severity of the problem can be seen from the findings of different surveys which are presented in Table 5.

As Table 5 shows, in many cases, offshoring projects have produced disappointing results. In some projects, cost savings are close to 50 percent whereas there are no cost savings at all in others. An insufficient cost analysis leads to a false estimation of the expected savings [69].

In order to prevent disappointing cost savings, the implementation of a realistic method to predict total costs by considering all cost categories is required [69]. Without a proper understanding of offshore complexity, hidden costs and risks will blindside IT decision makers and prevent the attainment of anticipated wage rate savings [70].

Table 6. Techniques to reduce the cost of IT offshoring

Technique	Description
Vendor contract (in the case of outsourcing)	A contract which includes Service-Level Agreements (SLAs) signed off by all parties, including business users. Vendors must demonstrate the operation of defined processes for all work identified in the contract and how they will ensure they are followed.
Industry standard	Consistent and appropriate use of industry standards like Capability Maturity Model Integration (CMMI).
Achievable project plan	A comprehensive and achievable project plan for transitioning support services from the current provider to the vendors support team.
Knowledge notebook	Detailed application knowledge notebooks captured by the vendor from the current application experts, including the current team leader.
Vender training program	In order to communicate with application users, a training program defining the companys business functions and terminology must be developed for the vendors team.
Verifiable training process	A verifiable process for cross-training and re-training vendor support staff on critical application functions, problem history, and business knowledge.
Communication infrastructure	Communication procedures, infrastructure and protocols in place where appropriate, video/audio conferencing facilities.
Performance report	Work status, resource utilization, SLA performance metrics report, schedules and recipients clearly defined. Customer satisfaction reports should be based on actual work performed not periodic and anecdotal customer satisfaction surveys.
Data collection	A data collection tool must be used throughout the vendors entire support team that captures data on all support activities: work type, status, scheduling and backlog, resource loading, time tracking and data required for accurate SLA performance reporting.
QA reviews	Quality Assurance (QA) processes and reporting in place for both the transition of services and continued support operations. Operational QA reviews are for the life of the contract and are usually quarterly unless problems demand more frequent back-checks.
Process improvement programs	Process improvement programs should be an outcome of the QA process and agreed by both the vendor, IT sponsor and business customer.

3.3. Q3. What effort estimation models for GSD are available in the literature?

Due to the increasing tendency to utilize GSD in the last few decades, researchers have focused on effort estimation in GSD. This paper reviews the most up-to-date research published in this field. GSD effort estimation is quite different to collocated software development as GSD consists of additional cost factors. Despite its brief history, a considerable amount of research has already been directed towards effort estimation in GSD by introducing new models, identifying the unsuitability of existing collocated models and raising the problem of not having a proper model. This demonstrates the significance of the problem. However, only a handful of effort models have been introduced to predict the effort required for GSD. In fact, most of this research is still in the initial stage.

The identified models can be divided into two categories, based on the approach used for their development, as shown in Figure 3. A detailed discussion on these models follows with the authors critical review.

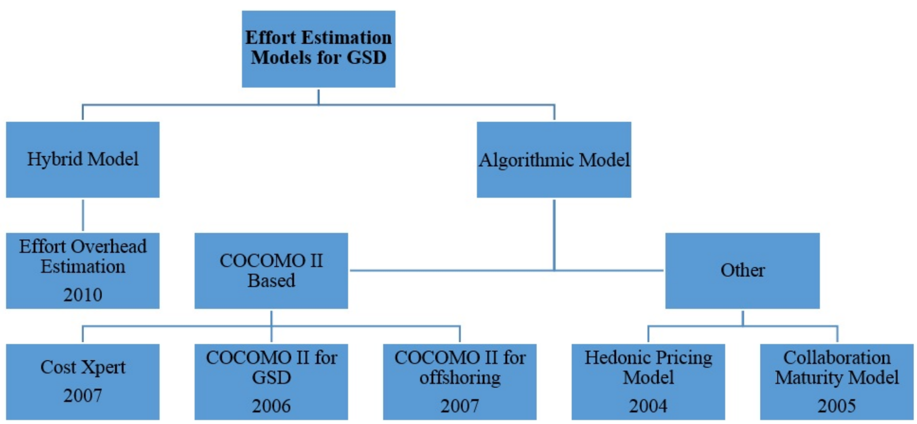

Fig. 3. Existing Effort Estimation Models for GSD

Figure 3 shows the name of the model and also which year it was introduced to illustrate the research trends on effort estimation in GSD over the last decade. All these models were introduced after 2004.

Effort Overhead Estimation -2010: The most recent work published to date (Nov 2010) is that of Lamersdorf, Fernndez-del Viso Torre et al. in [72]. Their model was designed specifically for developing individual cost models using expert estimations and project data. The initial development of the model is based on a Spanish GSD organisation which is the premier IT company in Spain and a leading IT multinational in Europe. Cost drivers for the specific distributed development context are identified and a causal model is built by conducting interviews with six of the most experienced practitioners in the company.

The model was developed using CoBRA (Cost estimation, Benchmarking and Risk Analysis Model) which combines expert-driven and data-driven cost estimation methods.

This method can be used in organisations where only a small amount of data is available. The CoBRA process comprises the following five steps.

1. Collect possible cost drivers
2. Rank and select cost drivers
3. Build causal model
4. Quantify causal relationships
5. Analyse past projects

The initial cost drivers for step 1 were collected based on the literature search and the results of previous empirical studies. The main focus of the collection of possible cost drivers was the identification of factors causing overheads in distributed development and other factors were disregarded. The selection of the factors and the determination of their causal relationship was done via group discussion. Afterwards, each expert estimated the quantitative impact of each factor on cost overhead. Using the data from past projects, the nominal productivity is then computed. Practitioners were asked to rank the selected cost drivers. These results were then aggregated into one list of factors and formed a causal model.

Unfortunately, steps 4 and 5 have not been completed and the model is yet to be validated. This will be done in future work. The unavailability of project data was the major obstacle for validation. The strengths and weaknesses of the model are presented in Table 7.

Table 7. Strengths and weaknesses of Effort Overhead Estimation

Strengths	Weaknesses
Model is based on hybrid method (data driven and expert driven)	Results represent one specific environment and cannot be easily generalized to other contexts and organizations.
List of factors in causal model can generally be indicators for important effort drivers in GSD	The resulting model is based on interviewees rankings of the selected factors. It shows high standard deviation in most of the quantification.
Site-specific effort drivers are included	External validity is threatened by the fact that all interviewees are employed in one organisation.
Effort estimations do not solely rely on individual expertise as it is collected and stored for sharing among many people.	Study is restricted to projects with one remote site.
	No validation of the model. (more empirical studies are needed to validate this model)
	No use of any statistical methods for data analysis
	Time differences, currency differences and coordination implications have not been considered.

The selected data does not reflect the true GSD overhead cost factors such as time differences, currency differences and coordination implications. As this model is based on one company, it may not be applicable to other companies. It is recommended that a similar study be conducted with a large number of data sources and expertise from different companies around the globe.

COCOMO II Based Models: The Constructive Cost Model (COCOMO) II is the most widely used software development cost estimation model for collocated software development [73]. It is accepted internationally in organisations of all sizes [73]. The standard COCOMO 11 approach does not address the characteristic of distributed development in great detail. There is only multisite development cost driver linked to geographical distributed software development. As this is the only factor related to distributed development, it cannot reflect the inherent complexity and various overhead drivers for GSD. Therefore, it has been accepted that this model is not sufficient for effort estimation in GSD and many researchers have attempted to extend the COCOMO II model for the GSD environment.

Cost Xpert-2007: Cost Xpert is a different approach introduced by Madachy using COCOMO II for distributed environments and has been adapted widely. As distributed sites in a GSD project might have different characteristics, this model suggests effort multipliers for every site. Cost Xpert is different to the other models as its work allocation is based on phase rather than module or specific function. It allows different calendars for multiple teams in a project. [74]. This model has been developed with the support of the University of Southern California (USC) Centre for Systems and Software Engineering to better estimate globally distributed projects.

The new model adapts traditional cost estimation formulas for distributed teams by using phase-sensitive effort multipliers. A project can be defined in terms of the distribution of software work by phase per team. The unique attributes of each team are also used in the calculations for more detailed and accurate estimation. The model leverages the phase-sensitivity of effort multipliers to capture the variance due to different team characteristics by phase [74].

Examples of major global companies helping with the model include Unisys, Wipro Technologies and Cognizant Technologies. Companies are using the model in its current spreadsheet form (until it is included in a future Cost Xpert product update). Data is being collected for further model validation and local calibrations. The strengths and weaknesses of the model are presented in Table 8.

This model is attractive as it considers cost factors related to people in different teams (labour calendar, labour categories, rates with local currencies) at different phases for the calculations. But collaboration and coordination which has a major impact in the global environment has not been considered for the estimation and additional effort drivers are not introduced.

Refinement to COCOMO II for GSD-2006: P. Keil et al. [73] proposed an extension of the most widely used COCOMO II model for the distributed environment. This model introduced additional cost drivers for GSD. The multisite development cost driver in COCOMO II is determined by multisite collocation and multisite communication. In this method, the relevant drivers of these two factors and the additional factors that are related to GSD have been identified [73]. To tailor the COCOMO II framework, a list of

Table 8. Strengths and weaknesses of Cost Xpert

Strengths	Weaknesses
Work allocation by phase vs module	Model may not be suitable for systems where work allocations are based on modules
Different working calendars for each team is possible	Collaboration and coordination has not been considered
Effort multipliers are phase sensitive	No validation
Industry collaboration	No quantification
	Additional cost drivers to COCOMO 11 are not introduced

complexity factors resulting from geographic distribution have been identified as follows, similar to the main classification of the effort multiplier cost drivers in COCOMO 11 [73]. These complexity factors below have been identified in several projects and are presented in different contributions (Siemens project):

– Product factors: Novelty of the software to be developed, architectural adequacy
– Personnel factors: Cultural fit, skill level, shared understanding, information sharing constraints
– Project Factors: Novelty of collaboration model, tools and infrastructure, physical distance

The objective of this research was to provide a decision-making framework for managers to calculate the trade-off between the collocated and distributed development of a software product. Evidence is not available to show whether the objective was achieved or not. Even though it suggests a set of new effort multipliers that consider the impact of distributed collaboration, it does not quantify the impact with a justification.

It is stated that further work for this research requires the verification and improvement of cost factors, calibrating the relevance of each project factor and refining the COCOMO II model for global software cost estimation [73]. It seems this is still at a very early stage and needs further research. Project data have been collected from globally distributed projects to increase the understandability of the system. They observed from the data that there was not only a cost associated with communication among sites but there was also an organizational cost associated with staff.

Table 9. Strengths and weaknesses of Refinement to COCOMO II for GSD

Strengths	Weaknesses
Complexity factors resulting from geographical distribution have been identified and nine new effort drivers are introduced	No quantification of the identified complexity factors
Additional multipliers are categorised based on the original category in COCOMO II	No validation
	Does not appear to be a systematic approach to identifying factors
	Factors have been derived from the authors experience and other publications

COCOMO II for Offshoring-2007: The most comprehensive approach published to date on estimating the effort in distributed software projects is the extension to the well-known COCOMO II by Betz and Mki [75]. They focused on the widely used post-architecture model in COCOMO 11 and undertook the amplification of COCOMO II has been carried out in three steps. They started with the identification of additional cost drivers for distributed development, categorizing them into different groups and assigning a value for each group/driver [75].

The existing cost drivers of COCOMO II were analyzed and their relevance to global software development was identified [75]. Additional cost drivers for offshore outsourcing software development were identified by conducting a qualitative survey based on semi-structured interviews with German software producers [75]. Newly introduced factors identified in the interviews were categorized based on theoretical thought, the research in the literature and expert opinion.

Eleven new effort multipliers were added to the equation and these were termed Effort Multipliers Outsourcing (EMO) [75]. These factors were categorized into four groups: Outsourcing Factors, Buyers Outsourcing Maturity, Providers Outsourcing Maturity and Coordination Factors [75] which are presented in Table 10.

Table 10. Effort Multipliers Outsourcing (EMO) [75]

Outsourcing Factors	Buyers Outsourcing Maturity	Providers Outsourcing Maturity	Coordination Factors
CULT	BOXP	OOXP	OFIT
BALA	BUPM	PUPM	PMGM
TMZN	CODS	-	TESP

Outsourcing Factors: defines three basic and static cost drivers which may arise when collaborating with an international partner. The cost drivers are: cultural distance (CULT), barrier of language (BALA), and the different time zones (TMZN). The capabilities of the employees are not quantified by these factors because they are very abstract [75].

Buyers Outsourcing Maturity: defines cost drivers which specify the offshore outsourcing maturity of the buyer. Three factors are critical: buyers outsourcing experience (BOXP), buyers project managers (BUPM), and contract design (CODS). The BOXP refers to the actual experience of the buyer with offshore outsourcing projects. The BUPM evaluates the capabilities of the buyers project manager in comparison with their offshore qualifications. The CODS refers to the complexity of a collaboration contract [75].

Providers Outsourcing Maturity: defines cost drivers which specify the offshore outsourcing providers maturity. Two factors can influence the effort: providers outsourcing experience (POXP) and providers project managers (PUPM). The POXP refers to the actual experience of the provider with offshore outsourcing projects. The PUPM evaluates the capabilities of the providers project manager considering their offshore qualifications [75].

Coordination Factors: are cost drivers originating from the interaction between two partners. The additional effort is represented by three factors: outsourcers fit (OFIT), project management (PMGM), and team spirit (TESP). The OFIT refers to the right selection of a particular partner. The wrong partner increases the effort. The PMGM refers to the increased effort which is inherent in any offshore outsourcing project. The TESP also decreases the possible effort by raising the team spirit through team building meetings, common goals, and a mixture of off- and onshore team members [75].

According to the COCOMO 11 categories and values, additional factors were categorised and quantified with numerical values as the second and third phase of the amplification process.

The estimated result of the amplified COCOMO II model shows that the effect of offshore outsourcing software development projects increases by at least 50 percent and at the most by 750 percent in the worst case scenario [75]. But it also indicates that there can still be a cost advantage as the wage level in some countries is eight times higher than in others [75].

They are aware of the lack of validation because of missing data on actual offshore outsourcing software development projects. But they claim that they are confident that this approach is heading in the right direction and it needs to be further calibrated.

It should be noted that this is still a work in progress and the model needs more calibration and validation [75]. This model has been simplified by reducing the number of collaborative companies to two, which is not always recommended in the context of GSD.

Table 11. Strengths and weaknesses of COCOMO II for Offshoring

Strengths	Weaknesses
Most comprehensive models published for GSD in the literature.	Model has been simplified by restricting collaboration companies to two.
Follows a systematic approach compared to other models published so far.	Lack of validation because of missing data of actual offshoring software development.
11 new effort drivers are introduced to COCOMO 11.	Quantification has not been conducted in a systematic way and the derivation of numerical values is not very clear.

Comparison of two similar modules with COCOMO 11: Several factors in COCOMO 11 are affected by GSD which encouraged the researchers in [75] to consider several factors from the existing model. Table 12 provides more detail.

Analysis of the Models: A comparison of these models with respect to the main focus, the modelling techniques used, the contribution of the model and other aspects considered at the formulation of the model are shown in Table 13.

Table 12. Comparison of two extensions to COCOMO II

COCOMO II Effort Drivers	COCOMO II for GSD	COCOMO II for Offshoring
Product (5)	Not considered	1 document match to life cycle needs
Platform (3)	Not considered	Not considered
Personnel (6)	Not considered	All personnel
Project (3)	1multisite development	1multisite development
New Effort Drivers	9 new drivers were introduced	11 new drivers were introduced

Table 13. Comparison of existing effort estimation models

Factor	Effort Overhead Estimation	COCOMO II for GSD	COCOMO II for offshore outsourcing	Cost Xpert 2007
Contributions	Important factors for a particular company	No clear outcome	Effort is higher in outsourcing	Consideration of dissimilar attributes at each site
Main focus	GSD effort overhead	Offshore insourcing	Offshore outsourcing	Distributed global development
Modelling technique	Hybrid	Algorithmic	Algorithmic	Algorithmic
Development method	COBRA	COCOMO II	COCOMO II	COCOMO II
Software size measure	Not considered	KLOC, Function point	KLOC, Function point	KLOC, Function point
Estimation method	Qualitative and quantitative	Quantitative	Quantitative	Quantitative
Number of projects used for the study	1	Based on the literature	Several projects	4
Number of remote sites supported	2	Not given	2	More than 2
Quantification/ calibration	Done partially	Not done	Done partially	Used existing model (N/A)
Validation	Not done	Not done	Done partially	Not done
Evidence of industry collaboration	Yes	No	No	Yes
Site-specific effort drivers	Only general	Only general	Only general	Considered specific
Site-dependent impact	Yes	Yes	Yes	No
Task-dependent impact	Yes	Yes	No	No

Summary of the findings: The aforementioned discussion highlights the related work on effort estimation in GSD. According to the information presented in Table 13, the following observations can be made.

- Most of the models are based on quantitative algorithmic techniques and 75 percent of them are based on the widely used COCOMO 11.
- None of the models performs proper calibration so they are not in a form which is suitable for use.
- The models have not been empirically validated.
- None of the models addresses the two main GSD-specific characteristics sufficiently. Only the CostXpert model considers dissimilar attributes in each site but it does not have new GSD effort drivers to measure communication and coordination overhead in a distributed environment. The other two extensions to COCOMO 11 have new effort drivers to measure additional effort but they do not consider dissimilar attributes in each location.
- All the existing studies on GSD effort overhead calculations are limited to two remote sites and therefore, complexity in multi-site interaction has received limited attention.
- Site-dependent parameters have been considered to an acceptable level whereas task-dependent parameters have received limited attention.
- All these models are still at an early stage and further research is required to improve them and make them suitable for use in practice.

Overall, the above review of the existing literature related to GSD effort estimation methods has enabled the identification of the following knowledge gaps which need to be addressed in future research.

- The effort model should address both dissimilar attributes at each site and GSD overheads related to collaboration.
- Effort versus cost in GSD: The existing models mainly aim to calculate effort only. In the literature on collocated effort estimation, no difference is usually made between effort and cost. This is mainly because all costs in software development are personnel costs, which depend directly on effort. However, in GSD, cost rates differ between sites. Therefore, effort and cost cannot be used as synonyms in the GSD context and the estimation of effort is not sufficient for decision making, hence a cost estimation model is also required.
- The evaluation of GSD overhead in multiple interactions: The existing suggestions are limited to overhead calculation when only two sites are connected. When the distributed site has multiple interactions, additional impact analysis is not straightforward and a standard procedure is needed. Figure 4 shows an example of five different sites to illustrate the increased complexity.
- The following concerns are noted in relation to the calculation of GSD overhead at each site:
  - The calculation of additional effort for S3, S4 and S5 is quite straightforward as they interact with only one other site.
  - The calculation of additional effort at S1 and S2 is more complex as they interact with more than one site. GSD-specific effort drivers for S1 have an impact on [S1-S2], [S1-S4] and [S1-S5]. Taking the average of all three may hide the true picture and is not an accurate solution. Therefore, the impact of all needs to be taken into consideration, however, as most of the attributes are qualitative

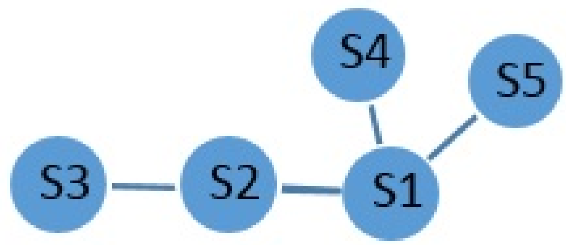

Fig. 4. Interaction between geographical locations

in nature (language differences, cultural differences etc.), these values cannot be added together. It is necessary to devise a standard scale which considers both the interaction values and the number of interactions. To obtain the interaction value, work dependency must be known.

– Model calibration and validation: The scarcity of GSD project data in the public domain prevents the calibration and validation of the models.
– The literature review indicates that communication and coordination impacts are unpredictable if the work dependency among sites is unknown. Therefore, the lack of a systematic work allocation method has become a major hurdle for accurate effort estimation in GSD.

Q4. What systematic studies have been conducted on effort estimation in GSD?

No systematic literature surveys have been conducted on effort estimation models in GSD whereas surveys have been conducted on cost estimation approaches [27], systematic reviews [76] and software estimation techniques for traditional collocated software development [77]. Our study reflects that this is a high time to conduct a proper systematic review on effort estimation in GSD. This is evidenced by the large number of research papers recently published in this discipline which create the requirement of having a systematic arrangement of published research work for future researchers quick reference.

At the International Conference on Global Software Engineering 2010, the preliminary results from a survey conducted in a multinational organization on the application of estimation techniques in GSD projects [78] was presented. The survey was carried out using an online questionnaire to record the opinions of people in the selected organization which has 3595 employees in the IT department. Recommendations were given based on the 551 answers received from the employees in the IT department. According to the survey results, estimation techniques used for GSD applications in this company are Delphi, Function points, Planning Poker, Usecase points, Expert judgment and Historical data. Some of the employees were not aware of the techniques used for estimation.

The results indicate that the current set of effort estimation techniques and models have been used without a clear idea of what needs to be different when such techniques or models are used in the context of GSD. In addition, this indicates that there is no correlation between the effort estimation techniques being used and the number of locations involved in the project. Further, it can be seen that although different kinds of effort estimation techniques are used by GSD projects, none of them is considered a proper model for the distributed global environment.

4. Research Directions and Future Challenges

The systematic study conducted on effort estimation in GSD clearly indicates that effort estimation has become a prominent research problem in the software engineering discipline. Many researchers have worked in this area, some since the early 2000s. Our survey shows the importance of effort estimation and the contribution of the knowledge of many researchers to enhance the immature area of effort estimation in GSD. In this section, we identify the unsolved challenges in this area. In particular, different research directions in relation to effort estimation in GSD are discussed.

Research Direction: Answering the questions in section 3 will open an avenue for new research challenges. A new question that can be formulated is as follows:

How should the effort of a distributed offshore software development/GSD project be estimated in order to increase the chance of success?

The previous discussion provides a foundation and basic information on the complexity of GSD and the available methods in the literature for effort estimation. According to the definition of project success, if the project can be completed within the planned budget while achieving other goals such as time, scope and quality, the project will be successful. By introducing a proper model to estimate the development cost, the success rate of a globally distributed project can be improved.

How can a negative influence on the success of offshore development projects be quantified? If organizations can find an answer to this question, they will be able to select the best offshoring partner who has a less negative impact before starting the project. This will solve the problem to a certain level. This solution has become a viable one as many potential offshoring partners are available all over the world with attractive labour costs at first sight. However, it has been realized that attractive cheap labour does not show the true picture. Hence, many researchers have already highlighted the significance of the problem and the importance of effort estimation in GSD.

Balancing the success factors and coordination measures: In the introduction, we mentioned that the main reason for offshoring is the reduction of software development costs. The expectation of cost reduction is mostly not realized [80]. The main reason for this is that not all costs involved in the process of GSD are included in the cost estimation. The expected cost reduction is mostly based on lower wages in other countries. The coordination measures should be added into an offshore software development project. This costs extra time and money. Sakthivel (2005) [3] mentions sourcing, contracting, project management and risk estimation costs as part of this increase in costs.

In short, the problem we have is: does offshoring save money? To answer this question, the following points can be made:

1. Offshoring allows cheap labour
2. Offshoring requires quality communication

3. Quality communication requires proper coordination measures
4. Coordination measures cost money

1 is less than OR greater than 4 decides whether it is worthwhile to go offshore.

Our analysis indicates that all the models proposed by researchers are still in their early stages. A detailed discussion with a critical review is given in section 3, which shows that the communication and coordination which has a great impact on the GSD effort has not been considered in the proposed models. All the models proposed to date mainly focus on traditional software development in a global environment using collocated models which have been extended. Most of these models were developed in the seventies and eighties. Thus, it is doubtful that the foundations of these models are applicable to todays technology. But when we observe the technological advancements of the software industry, component-based software development, service-oriented architecture, open-source software development and enterprise resource planning are the widely used emerging technologies. Component-based software development (CBSD) is more suitable for GSD as it can easily recombine and reuse new products [82]. It has been suggested that CBSD will improve GSD practices by allowing each site to take ownership of particular components, which in turn, will result in reduced inter-site communication and coordination activities [83]. Hence, scholars should focus on developing effort estimation models for globally distributed development using the emerging technologies.

Although, none of the models proposed so far is sufficient to estimate the required effort for GSD using the emerging technologies, these models have made a valuable contribution for future research with their findings. For example, task allocation methods, the impact of collaboration maturity, suitable contract types based on the type of project and the identification of cost drivers in GSD are some of the significant contributions to future research in this area.

Success in terms of scope, quality, time and costs is dependent on the choice of coordination measures. A firm has to choose the right coordination measures in order to achieve success. If costs are important and time and quality are less important, less coordination measures are required. If time or quality is important, the offshore project needs extra coordination measures. This was also noted by Harmsen, Lubbers and Wijers (1995) [81]

5. Conclusions

In this paper, we have presented a systematic review of effort estimation in GSD with a comprehensive discussion on the models available. We have discussed its importance, benefits and challenges. The existing models for effort estimation in GSD in the literature have been analysed in terms of the modelling techniques used, estimation types, data required and level of acceptability based on validation. Future challenges have been identified with suggested research directions. Our systematic study shows that reduced development cost is one of the major motivations for GSD however our study reveals the lack of models to estimate the true costs. Hence, additional managerial overhead, additional time to build up coordination, travel expenses and restrictions on travel in order to limit costs can lead to problems such as a lack of trust which can lead to a reduction in productivity. Finally, we identified several research directions and future challenges which need the attention of future researchers.

It is very challenging and almost impossible to propose a model suitable for use across the globe as required by the nature of GSD. Since GSD scenarios are diverse, what works in one context might not directly apply in another. Hence, a one-fits-all solution is not applicable and will not be of much use to anyone. Thus, we need to devise a model which is customisable, based on the factors which are relevant to where the software is developed. Hence, our next step is to select the most widely used technology for global software development and design a research framework to develop an estimation model for the selected technology, incorporating additional the cost drivers involved in GSD. As the design specification is the input for most software estimation models, it is necessary to investigate how a design specification (software architecture) differs when software is developed in a globally distributed environment.

Table 14. Issues in software effort estimation

Issue	Description
Definition	Lack of clear and accepted definitions for drivers, such as size, quality, complexity, experience, etc.
Quantification	The majority of the cost drivers are hard to quantify. Often one has to use measures such as many, moderate, few, etc.
Objectivity Subjectivity	This is a potential risk factor. What may be complex for developer A may not be complex for developer B.
Correlation	It is difficult to consider one driver by itself. A change in the value of driver A may have consequences for the values of several other cost drivers. This makes measurability difficult.
Relation between driver and effort	For estimation, it is important to predict the relation between, for example, software size and the required effort, a specified quality level and required effort, etc. From the literature, we know that there is little clarity around these relations.
Calibration	It is impossible to talk about ”the most important” cost drivers in isolation. It differs from situation to situation.
Effectiveness and efficiency	There is conflict between effectiveness and efficiency. From the effectiveness perspective, it is worthwhile to pay a lot of attention to, for example, user participation. For the efficiency of a project, it is justifiable to avoid user involvement.
Human factors	Almost all research agrees on the dominating influence of cost drivers, such as experience and quality of the personnel. This means that investment in ’good’ developers is important.
Reuse	In many studies, reuse is regarded as (one of) the most important ways to increase productivity.

Once the new model is developed, mathematical and empirical validation will be undertaken to determine the viability of the model. As the area matures, an increased focus on empirically supported results leads to a greater potential impact on future research and industrial practice. Since GSD does not have a standard modelling mechanism, it is hard to find the required empirical data for validation. Further, as it contains sensitive information, if the data is available it might be difficult to get access to the available data as a research student. Thus, it is necessary that GSE-related empirical findings be reported so as to be useful for practitioners and researchers. Furthermore, it is important to summarize progress in order to see the big picture of the published research to identify gaps and commonalities [19].

In addition to the above challenges, there are several common issues which apply to estimation models. In estimation, it is necessary to identify the most dominant cost factors. When making an estimation, one has to identify which cost drivers are the most important for the specific situation, what the values are for the cost drivers and what influences the effort. There are several issues that need special attention in software effort estimation which are presented in Table 14.

Acknowledgments. A La Trobe University postgraduate scholarship and a write-up awards which supported this work are hereby acknowledged by the first author.

References

1. P. Taylor, D. Greer, P. Sage, G. Coleman, K. McDaid, and F. Keenan, ”Do agile GSD experience reports help the practitioner?,” Proceedings of the international workshop on Global software development for the practitioner, pp.87-93, 2006.
2. J. Herbsleb and D. Moitra, ”Global software development,” Software, IEEE, vol. 18, pp. 16-20, 2002.
3. S. Sakthivel, ”Virtual workgroups in offshore systems development,” Information and Software Technology, vol. 47, pp. 305-318, 2005.
4. D. mite and J. Borzovs, ”Managing Uncertainty in Globally Distributed Software Development Projects,” Datorzin tne un inform cijas tehnolo ijas, p. 9, 2008.
5. R. Prikladnicki and J. Audy, ”Process models in the practice of distributed software development: A systematic review of the literature,” Information and Software Technology, 2010.
6. M. Maznevski and K. Chudoba, ”Bridging space over time: Global virtual team dynamics and effectiveness,” Organization science, vol. 11, pp. 473-492, 2000.
7. D. Smite, ”Project Outcome Predictions: Risk Barometer Based on Historical Data,” in International Conference on Global Software Engineering(ICGSE 2007), 2007, pp. 103-112.
8. M. Fabriek, M. van den Brand, S. Brinkkemper, F. Harmsen, and R. Helms, ”Improving offshore communication by choosing the right coordination strategy,” Citeseer2007.
9. E. Conchir, P. gerfalk, H. Olsson, and B. Fitzgerald, ”Global software development: where are the benefits?,” Communications of the ACM, vol. 52, pp. 127-131, 2009.
10. E. Conchuir, H. Holmstrom, P. Agerfalk, and B. Fitzgerald, ”Exploring the assumed benefits of global software development,” in International Conference on Global Software Engineering(ICGSE 2006) 2006, pp. 159-168.
11. C. Ebert and P. De Neve, ”Surviving global software development,” Software, IEEE, vol. 18, pp. 62-69, 2001.
12. G. Seshagiri, ”Global Software Development-Point/Counterpoint-GSD: Not a Business Necessity, but a March of Folly,” IEEE Software, vol. 23, pp. 62-65, 2006.
13. W. DeLone, J. Espinosa, G. Lee, and E. Carmel, ”Bridging global boundaries for IS project success,” Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

14. R. Aron and J. Singh, "Getting offshoring right," Harvard business review, vol. 83, p. 135, 2005.
15. N. Huda, N. Nahar, J. Tepandi, and P. S. Deo, "Key barriers for global software product development organizations," in Management of Engineering and Technology, 2009. PICMET 2009. Portland International Conference on, 2009, pp. 1081-1087.
16. P. gerfalk, B. Fitzgerald, H. Holmstrm Olsson, and E. Conchir, "Benefits of global software development: The known and unknown," Making Globally Distributed Software Development a Success Story, pp. 1-9, 2008.
17. D. Damian and D. Moitra, "Guest Editors' Introduction: Global Software Development: How Far Have We Come?," Software, IEEE, vol. 23, pp. 17-19, 2006.
18. E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," Software, IEEE, vol. 18, pp. 22-29, 2001.
19. D. mite, C. Wohlin, R. Feldt, and T. Gorschek, "Reporting Empirical Research in Global Software Engineering: a Classification Scheme," 2008.
20. M. Robinson and R. Kalakota, "Offshore outsourcing: Business models, ROI and best practices," Atlanta, GA: Mivar Press, Inc, 2004.
21. J. Miguel, "Challenges and improvements in distributed software development: A systematic review," Advances in Software Engineering, vol. 2009, 2009.
22. N. Moe and D. mite, "Understanding lacking trust in global software teams: A multi-case study," Product-Focused Software Process Improvement, pp. 20-34, 2007.
23. R. Gonzalez, J. Gasco, and J. Llopis, "Information systems outsourcing: A literature analysis," Information and management, vol. 43, pp. 821-834, 2006.
24. A. Avritzer, D. Paulish, Y. Cai, and K. Sethi, "Coordination implications of software architecture in a global software development project," Journal of systems and software, 2010.
25. B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Engineering, vol. 2, 2007.
26. B. Kitchenham, "Procedures for performing systematic reviews," Keele, UK, Keele University, vol. 33, 2004.
27. B. Boehm, C. Abts, and S. Chulani, "Software development cost estimation approachesA survey," Annals of software engineering, vol. 10, pp. 177-205, 2000.
28. M. Shepperd, "Software project economics: a roadmap," in Proceedings of the International Conference on Software Engineering - Future of Software Engineering (FOSE '07), 2007, pp. 304-315.
29. S. Sharma, "Global Software Development-Point/Counterpoint-Making Global Software Development Work," IEEE Software, vol. 23, p. 62, 2006.
30. J. D. Herbsleb and R. E. Grinter, "Splitting the organization and integrating the code: Conway's law revisited," in Software Engineering, 1999. Proceedings of the 1999 International Conference on, 1999, pp. 85-95.
31. J. Espinosa and E. Carmel, "The effect of time separation on coordination costs in global software teams: A dyad model," in Proceeding of the 37th Hawaii International Conference on System Sciences, 2004, p. 10.
32. A. Delmonte and R. McCARTHY, "Offshore software development: Is the benefit worth the risk," 2003, pp. 16071613.
33. J. Dibbern, T. Goles, R. Hirschheim, and B. Jayatilaka, "Information systems outsourcing: a survey and analysis of the literature," ACM SIGMIS Database, vol. 35, pp. 6-102, 2004.

34. A. Almeida, R. Souza, G. Aquino, and S. Meira, "Effort Drivers Estimation for Brazilian Geographically Distributed Software Development," Software Engineering Approaches for Offshore and Outsourced Development, pp. 60-65, 2009.
35. A. Khan and Z. Muhammad, "Exploring the Accuracy of Existing Effort Estimation Methods for Distributed Software," 2009.
36. K. Chan and L. Chung, "Integrating process and project management for multi-site software development," Annals of software engineering, vol. 14, pp. 115-143, 2002.
37. D. Damian, F. Lanubile, and H. Oppenheimer, "Addressing the challenges of software industry globalization: the workshop on Global Software Development," in ICSE '03 Proceedings of the 25th International Conference on Software Engineering, 2003, pp. 793-794.
38. D. Farrell, "Smarter offshoring," Harvard business review, vol. 84, p. 84, 2006.
39. G. Davis, P. Ein-Dor, W. King, and R. Torkzadeh, "IT offshoring: History, prospects and challenges," Journal of the Association for Information Systems, vol. 7, pp. 770-795, 2006.
40. H. Holmstrom, E. Conchuir, P. Agerfalk, and B. Fitzgerald, "Global software development challenges: A case study on temporal, geographical and socio-cultural distance," 2006.
41. J. Herbsleb, A. Mockus, T. Finholt, and R. Grinter, "Distance, dependencies, and delay in a global collaboration," 2000, pp. 319-328.
42. M. Bass and D. Paulish, "Global software development process research at Siemens," in The 3rd International Workshop on Global Software Development, 2004, pp. 11-14.
43. R. Grinter, J. Herbsleb, and D. Perry, "The geography of coordination: dealing with distance in R and D work," 1999, pp. 306-315.
44. R. Ravichandran and N. Ahmed, "Offshore systems development," Information and management, vol. 24, pp. 33-40, 1993.
45. S. Krishna, S. Sahay, and G. Walsham, "Managing cross-cultural issues in global software outsourcing," Communications of the ACM, vol. 47, pp. 62-66, 2004.
46. W. Aspray, F. Mayadas, and M. Vardi, "Globalization and offshoring of software," Report of the ACM Job Migration Task Force, Association for Computing Machinery, 2006.
47. J. Herbsleb, A. Mockus, T. Finholt, and R. Grinter, "An empirical study of global software development: distance and speed," 2001, pp. 81-90.
48. A. Gopal, T. Mukhopadhyay, and M. Krishnan, "The role of software processes and communication in offshore software development," Communications of the ACM, vol. 45, pp. 193-200, 2002.
49. M. Jennex and O. Adelakun, "Success factors for offshore information system development," Journal of Information Technology Cases and Applications, vol. 5, pp. 12-31, 2003.
50. C. Cramton, "The mutual knowledge problem and its consequences for dispersed collaboration," Organization science, vol. 12, pp. 346-371, 2001.
51. J. Dibbern, J. Winkler, and A. Heinzl, "Explaining variations in client extra costs between software projects offshored to India," MIS quarterly, vol. 32, pp. 333-366, 2008.
52. D. E. Damian and D. Zowghi, "The impact of stakeholders' geographical distribution on managing requirements in a multi-site organization," in Requirements Engineering, 2002. Proceedings. IEEE Joint International Conference on, 2002, pp. 319-328.

53. J. Herbsleb, D. Paulish, and M. Bass, ”Global software development at siemens: experience from nine projects,” 2005, pp. 524-533.
54. P. gerfalk, B. Fitzgerald, H. Holmstrm, B. Lings, B. Lundell, and E. Conchir, ”A framework for considering opportunities and threats in distributed software development,” 2005, pp. 4761.
55. M. Rao, ”Key issues for global IT sourcing: Country and individual factors,” EDPACS, vol. 32, pp. 1-11, 2004.
56. J. Espinosa, N. Nan, and E. Carmel, ”Do gradations of time zone separation make a difference in performance? A first laboratory study,” 2007, pp. 12-22.
57. M. Muhairat, S. Aldaajeh, and R. Al-Qutaish, ”The Impact of Global Software Development Factors on Effort Estimation Methods,” European Journal of Scientific Research, vol. 46, pp. 221-232, 2010.
58. C. Ranganathan and S. Balaji, ”Critical capabilities for offshore outsourcing of information systems,” Working Papers on Information Systems, 2007.
59. B. Businessweek, ”The Hidden Cost of IT Outsourcing,” 2003.
60. S. Overby, ”The hidden costs of offshore outsourcing,” CIO-FRAMINGHAM MA-, vol. 16, pp. 60-66, 2003.
61. neoIT, ”RESEARCH SUMMARY: Total Cost of Offshore (TCO): Understanding The True Offshore Financial Rewards and Costs,” Offshore Insights White Paper Series, vol. Volume 2, Issue 4, 2004.
62. N. Gorla and Y. Lam, ”Who should work with whom?: building effective software project teams,” Communications of the ACM, vol. 47, pp. 79-82, 2004.
63. P. Pendharkar and J. Rodger, ”The relationship between software development team size and software development cost,” Communications of the ACM, vol. 52, pp. 141-144, 2009.
64. S. Faraj and L. Sproull, ”Coordinating expertise in software development teams,” Management Science, vol. 46, pp. 1554-1568, 2000.
65. R. Hirschheim and M. Lacity, ”The myths and realities of information technology insourcing,” Communications of the ACM, vol. 43, pp. 99-107, 2000.
66. M. Murray and R. Crandall, ”IT offshore outsourcing requires a project management approach,” SAM Advanced Management Journal, vol. 71, p. 4, 2006.
67. J. Barthelemy, ”The hidden costs of IT outsourcing,” Sloan Management Review, vol. 42, pp. 60-69, 2001.
68. D. Bradstreet, ”Dun and Bradstreet’s Barometer of Global Outsourcing,” ed. Business Wire: vol, 2000.
69. M. Amberg, G. Herold, R. Kodes, R. Kraus, and M. Wiener, ”IT Offshoring-A Cost-Oriented Analysis,” Manuscript for the CISTM,, 2005.
70. I. Computer Aid, ”The Challenges of Offshore Outsourcing: Understanding the Risks and Hidden Costs,” World Leaders in IT Process and Productivity, 2004.
71. C.W. L. I. I. P. a. Productivity, ”Use Field-Proven Techniques to Avoid the Hidden Costs of IT Outsourcing, Computer Aid News and Notes.”
72. A. Lamersdorf, A. Fernndez-del Viso Torre, J. Mnch, C. Snchez, and D. Rombach, ”Estimating the Effort Overhead in Global Software Development,” International Conference on Global Software Engineering(ICGSE 2010), 2010.
73. P. Keil, D. Paulish, and R. Sangwan, ”Cost estimation for global software development,” 2006, p. 10.

74. R. Madachy, ”Distributed global development parametric cost modeling,” Software Process Dynamics and Agility, pp. 159-168, 2007.
75. S. Betz and J. Mki, ”Amplification of the COCOMO II regarding Offshore Software Projects,” Offshoring of software development, p. 33, 2008.
76. M. Jorgensen and M. Shepperd, ”A systematic review of software development cost estimation studies,” Software Engineering, IEEE Transactions on, vol. 33, pp. 33-53, 2006.
77. M. Nasir, ”A Survey of Software Estimation Techniques and Project Planning Practices,” Proceedings of the Seventh ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and parallel!Distributed Computing, 2006.
78. C. Peixoto, J. Audy, and R. Prikladnicki, ”Effort Estimation in Global Software Development Projects,” International Conference on Global Software Engineering(ICGSE 2010), 2010.
79. E. Carmel and P. Tjia, Offshoring information technology: sourcing and outsourcing to a global workforce: Cambridge Univ Pr, 2005.
80. R. Kuni and N. Bhushan, ”IT application assessment model for global software development,” in International Conference on Global Software Engineering(ICGSE 2006), 2006, pp. 92-100.
81. F. Harmsen, I. Lubbers, and G. Wijers, ”SuccessDriven Selection of Fragments for Situational Methods: The S3 Model,” 1995, pp. 104-115.
82. J. Kotlarsky, I. Oshri, J. Hillegersberg, and K. Kumar, ”Globally distributed component-based software development: an exploratory study of knowledge management and work division,” Journal of Information Technology, vol. 22, pp. 161-173, 2007.
83. J. Kotlarsky, I. Oshri, J. van Hillegersberg, and K. Kumar, ”Globally distributed component-based software development: an exploratory study of knowledge management and work division,” Journal of Information Technology, vol. 22, pp. 161-173, 2007.

Dilani N.Wickramaarachchi received her PhD degree in Computer Science and Computer Engineering from La Trobe University (LTU), Australia. She holds the MSc degree in Computer Science from University of Colombo School of Computing, Sri Lanka and the BSc degree in Industrial Management and Information Technology from University of Kelaniya, Sri Lanka, She is a senior lecturer in the department of Industrial Management, Faculty of Science, University of Kelaniya. Her current research interest areas are global software development, software testing and software measurement.

Richard Lai is with the Department of Computer Science and Information Technology at La Trobe University, Australia. Prior to joining La Trobe, he has spent more than 10 years in the computer and communications industry. He was ranked as the worlds number one scholar in systems and software engineering consecutively for four years (1999-2002), according to an annual survey published in the Journal of Systems and Software. His current research interests include component-based software system, software measurement, requirements engineering, and global software development.

Received: February 29, 2016; Accepted: October 10, 2016.