University Mohammed V

Doctoral Thesis

Software Cost Estimation in Global Software Development Projects

Author:
Manal El Bajta

Supervisor:
Dr. Ali Idri

A thesis submitted in fulfilment of the requirements for the degree of Doctoral Thesis

in the

Software Project Management Research Group
ENSIAS

April 2019

Declaration of Authorship

I, Manal El Bajta, declare that this thesis titled, 'Software Cost Estimation in Global Software Development Projects' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: Manal EL BAJTA

"It is not the knowledge which should come to you, it is you who should come to the knowledge."

Imam Malik

UNIVERSITY MOHAMMED V

Abstract

National School of Computer Science and Systems Analysis
ENSIAS

Doctoral Thesis

Software Cost Estimation in Global Software Development Projects

by Manal El Bajta

Global Software Development has become an established software development paradigm that promises several benefits over collocated software development. However, project managers of such projects face the socio-cultural diversity and new challenges arising from the distribution of different stakeholders geographically and temporally. The purpose of this thesis is threefold: i) To investigate the intended approaches for Software Project Management activities that can counter the effect of stakeholders distribution and provide a list of these approaches, their benefits and limits; ii) To carry out a systematic review to identify the addressed software cost estimation activities and attributes for distributed project ; and iii) To gather the dimensions encountered by GSD projects in a comprehensive Software Cost Estimation framework that can be ultimately implemented as a taxonomy. To gather empirical evidence, Systematic Mapping studies, Systematic Literature Reviews and a survey of researchers and industry practitioners were conducted. The research study presented in this thesis was conducted by close cooperation between the Software Project Management Research Group of ENSIAS in Morocco and the Software Engineering Research Group of University of Murcia in Spain.

Acknowledgements

Firstly, my sincere gratitude goes to my advisor and mentor Professor Ali Idri for giving me the opportunity to be part of this rewarding adventure within the Software Project Management Research Group. His continuous support, his patience and immense knowledge helped me in all the time of research and writing of this thesis. I thank him for every moment he chose to be the strict advisor, his guidance and pertinent advice enabled me to go through the hardest part.

Many thanks go to all members of the Software Engineering Research Group of University of Murcia in Spain. My friends, Juan Manuel Carrillo de Gea, Oscar Sanchez Ramon, Francisco Javier Bermudez Ruiz and my supervisors, Professor Jose Luis Fernandez-Aleman, Professor Joaquin Nicolas Ros and Professor Ambrosio Toval for their permanent cooperation and warm welcoming.

I would like to thank my friends and colleagues, Sofia Ouhbi for accepting nothing less than excellence from me, Saad Yassir Chadli, for the unforgettable moments of sincere happiness and support, Lamyae Sardi for constantly cheering and backing me up, and all my colleagues of the Software Project Management Research group.

Last but not least, my deepest gratitude goes to my family. To my father who encouraged me and provided me with all the strength I needed; my mother, for her overwhelming love and support; my lovely sister Intissare and brother Ziad. Many thanks to my dear husband Hicham, for believing in me, struggling with me and making my life easier and happier.

To Adam, the JOY and BEAT of my heart!

Contents

List of Figures

List of Tables

Abbreviations

ACM	**A**ssociation (for) **C**omputing **M**achinery
CA	**C**ost **A**ttribute
CBR	**C**ase-**B**ased **R**easoning
CMMI	**C**apability **M**aturity **M**odel **I**ntegration
COCOMO	**CO**nstructive **CO**st **MO**del
CORE	**CO**mputing **R**esearch (and) **E**ducation
EBSE	**E**vidence-**B**ased **S**oftware **E**ngineering
EC	**E**xclusion **C**riteria
EVM	**E**arned **V**alue **M**anagement
GA	**G**enetic **A**lgorithms
GSD	**G**lobal **S**oftware **D**evelopment
GSE	**G**lobal **S**oftware **E**ngineering
IC	**I**nclusion **C**riteria
ICGSE	**I**nternational **C**onference on **G**lobal **S**oftware **E**ngineering
IEEE	**I**nstitute of **E**lectrical (and) **E**lectronics **E**ngineers
IQMF	**I**nformation **Q**uality **M**anagement **F**ramework
IS	**I**nformation **S**ystem
ISBSG	**I**nternational **S**oftware **B**enchmarking **S**tandards **G**roup
ISO	**I**nternational **O**rganisation (for) **S**tandardization
IT	**I**nformation **T**echnology
JCR	**J**ournal **C**itations **R**eport
KA	**K**nowledge **A**rea
LOC	**L**ine **O**f **C**odes
MQ	**M**apping **Q**uestion
PICO	**P**opulation **I**ntervention **C**omparison **O**utcome

PM	**P**roject **M**anagement
PMBOK	**P**roject **M**anagement **B**ody **O**f **K**nowledge
PMI	**P**roject **M**anagement **I**nstitute
POI	**P**opulation **O**f **I**nterest
QA	**Q**uality **A**ssessment
QAP	**Q**uadratic **A**ssignment **P**rocedure
RBV	**R**esource **B**ased **V**iew
RQ	**R**esearch **Q**uestion
SCE	**S**oftware **C**ost **E**stimation
SDLC	**S**oftware **D**evelopment **L**ife-**C**ycle
SE	**S**oftware **E**ngineering
SEI	**S**oftware **E**ngineering **I**nstitute
SLR	**S**ystematic **L**iterature **R**eview
SMS	**S**ystematic **M**apping **S**tudy
SPM	**S**oftware **P**roject **M**anagement
SWEBOK	**S**oft**w**are **E**ngineering **B**ody **O**f **K**nowledge
XP	e**X**treme **P**rogramming

To my family...

Chapter 1

Introduction

"If you set goals and go after them with all the determination you can muster, your gifts will take you places that will amaze you." Les Brown

The main goal of the research presented in this thesis is to enable software project managers to properly manage and control the global software development (GSD), and to investigate also more accurate and efficient cost estimation models. This thesis aims also to gain an insight into the challenges and factors that affect the cost estimation in GSD projects and to identify actions that can be made to moderate their effects. The main contributions are: (i) investigation of the state of research in the area of GSD project management through a Systematic Mapping Study (SMS) [1], (ii) investigation of the state of research in Software Cost Estimation and (iii) proposing a framework to identify cost attributes (CAs) and associated mitigation scheme related to software cost estimation in a GSD context. The framework was populated by means of Systematic Literature Review (SLR) Studies and approved through a survey of industry practitioners.

This introduction is structured as follows: the motivation behind this thesis is presented in Section 1.1, while the research areas of this thesis are identified in Section 1.2. The main research objectives are presented in Section 1.3, the research contributions are reported in Section 1.4, and the research approaches are presented in Section 1.5. The thesis outline is presented in Section 1.6.

1.1 Motivations

Over the last few decades, the globalization trend has become a steady and irreversible trend towards corporate globalization and Information Technology (IT) outsourcing in particular [2]. This trend has several aspects which influence the world in various ways, including the emergence of overall production/consumer markets, the emergence of worldwide financial markets, the spread of political sphere of interests, increase in the flows of information, and also emergence of new ways of developing information systems [3]. The style of information systems development was enormously affected by the globalization. Because of the impact of globalization, numerous information systems developed by geographically co-located developers are delegated to other companies in various locations. The globalization is referred as global software development (GSD) to reduce temporal, geographic, social, and cultural distance across different countries [4].

Companies across the globe are capitalizing on globalization and outsourcing Information Technology (IT) to achieve business competitiveness. GSD provides the unprecedented possibility to reduce development costs by moving parts of the development work to low-wage countries, leverage large pools of skilled labor by coordinating across distance, reduce time to market by managing resources in multiple time zones and Closer Proximity to Market [5]. Faced with this evolving trend, Global Software Engineering (GSE), IT outsourcing and many software companies engaged in GSD activities have shown growth rates of 10-20% per annum [6]. The interest in the topic has also increased to reach its peak in 2006, the year of the first edition of the International Conference on Global Software Engineering (ICGSE).

Recent ICGSE industry keynote mentioned that only 30 percent of all embedded software is developed in a global or distributed context. However, global development and sourcing is growing, and the number of quality deficiencies and callbacks across industries is expanding in parallel with this growth [6]. So, Investigate new approaches and methods to successfully manage global software projects has rapidly become a key need across industries and researches [7]. Globally distributed projects, in particular, projects' life cycle, project management and controlling activities must be taken into special consideration by companies. In order to successfully achieve the fixed goals of a project, project management must include multiple activities such as planning, controlling and monitoring. This thesis presents first an overview of the approaches used by the project managers of GSD projects, investigates on software cost estimation applied to GSD projects, provides an integrative framework intended to help them conduct cost estimation in a GSD context and finally validates the accuracy of research findings through a survey of industry practitioners.

1.2 Research areas

The studies included in this thesis are carried out through the intersection of numerous research areas. They are concerned with SPM approaches in general and software cost estimation in particular when dealing with GSD projects. The following subsections introduce and describe briefly the concepts and their use in this thesis.

Global Software Development

Global software development (GSD) is the process by which the dispersion of the team extends across national boundaries in a coordinated fashion including both outsourcing and distributed teams within the same organization that are disbursed in different countries [8, 9]. In particular, GSD is now a mainstream business phenomenon owing to the advantages that it may have over collocated software development [5, 10]. It allows organizations to overcome geographical distances, to benefit from access to a larger qualified resource pool, to reduce development costs, to provide an easy access to customers, and to allow 24/7 work shifts. In GSD context, all stakeholders involved in the development process including customers, vendors, managers, developers, users, and so on might be scattered at various geographical locations of the globe. This dispersion adds new challenges and best practices to the already Software Development Life-Cycle (SDLC) activities [11, 12].

Software Project Management

The international standard (ISO/IEC/IEEE 12207: 2017 "Systems and Software Engineering – Software Life-cycle Process" [13]) presents software project management (SPM) as the process of managing, planning, monitoring, controlling, and leading software project from inception to closure using a scientific and structured methodology. Thus, SPM is about creating structure, managing the project commitments and achieving agreed results. However, without the use of project management, managers and organizations would be faced with an unpredictable and chaotic environment over which they have little control. The software development process is mainly implemented to meet the requirements of the project and to ensured both the progress and success of the software project.

Software Cost Estimation

The software industry is competitive and cost is at the center of most companies' concerns [14]. Therefore, software cost estimation (SCE) is one of most important issues in the cycle of development, management decision, and in the quality of software project. Cost estimation is thus used as a process of assigning value and predicting the effort required to develop a software system [15]. In particular, estimation is a clear vision for the costs of software development that project's manager can identify challenges, problems, and complexity of software planning. SCE emerged as an important discipline that continues to be a source of problems for practitioners despite improvement efforts.

1.3 Research objectives

Interest in GSD is rapidly growing as the software industry is experiencing an increasing globalisation of business. In GSD, stakeholders from different national and organisational cultures in addition to time zones are involved in developing software. Performing software development in a globally distributed environment implies numerous management activities that require a well-coordinated effort from multiple organisational actors or units. SPM for GSD offers a system of procedures, practices, and technologies that address the management and measurement of software engineering. Therefore, managers must perform effective software cost estimation and have an understanding of how software cost attributes related to distributed project can ensure the project's success.

The first objective of this thesis is to provide a list of available approaches available for use in a GSD context and highlighting their characteristics. The second objective is to provide a framework that helps managers estimate the software cost when dealing with GSD projects. To achieve these goals, several research questions have to be answered.

1. What are the approaches used by GSD project managers? What is the state of the art on this particular topic? Which SPM activities for GSD were most frequently addressed? What are the benefits and limits of those approaches in GSD? These questions are answered in Chapter 3.

2. What cost estimation techniques are used for GSD projects? What is the state of the art on this particular topic? Which software cost estimation activities have been addressed by GSD research? Which cost drivers affect GSD projects? Which cost estimation performances have been obtained from GSD projects? These questions are answered in Chapter 4.

3. What are the relevant CAs associated with the software cost management in GSD projects? This question is answered in Chapter 5.

4. How do researchers and practitioners rank the CAs associated with the management of GSD projects? How are these CAs treated in global software projects context? These questions are answered in Chapter 6.

5. How do taxonomy support cost estimation for GSD projects? What dimensions are used to document important cost estimation-related information? These questions are answered in Chapter 7.

1.4 Research contributions

This study reveals an insight into the challenges encountered when engaging in GSD projects by exploring the use of approaches related on SPM activities in general and SCE in particular. First, it discovers trends and gaps in GSD research area and provides an exhaustive list of approaches supporting SPM activities as well as their benefits and limits . This contribution has been carried out using an SMS as a research approach and can be useful to researchers wishing to pursue new research lines and to practitioners engaging in GSD activities. Second, it gives an overview of software cost estimation techniques, activities and cost drivers for GSD projects. A framework for software CAs of GSD projects populated by the means of an SLR, categorized using Software Engineering Institute (SEI) taxonomy and validated through a survey of practitioners and researchers has been conducted.

1.5 Research approaches

This thesis gathers data from academic researchers and industry practitioners using three research approaches:

Systematic review

Kitchenham et al. [16] define systematic review in general as: "means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest". As with other disciplines, software engineering in general and GSD in particular, employ systematic reviews to [17]:

- Summarise existing proof concerning a practice;

- Identify where there are gaps in current research, in order to help determine where further investigation may be required;
- Help position new research activities;
- Examine how far a given hypothesis is supported or contradicted by the available empirical evidence.

Framework

In general, a framework represents a real or conceptual structure intended to serve as a guide or support for the building of something that expands the structure into something valuable. Researchers in the software field provide framework to (1) review and synthesize relevant background literature that describes the influences affecting software, and (2) present a recurring model that describe influences on software development processes.

Survey

A survey research is "a research approach used to gather information about the incidence and distribution of, and the relationships that exist between, variables in a predetermined population" [18]. Surveys are conducted to gather data related to attitudes, behaviors and the incidence of events. Survey can be conducted in different settings, by means of interviews or questionnaires to a target population.

1.6 Thesis outline

This thesis is organized as follows:

Chapter 2 provides a summary of the studies discussing software cost management involved in the development of distributed software projects. Then, it gives an overview of the research areas related to this thesis namely, GSD, SPM, and SCE. Finally, it describes the research approaches that have been used throughout this work.

Chapter 3 presents an overview of SPM approaches used in GSD context. An SMS has been conducted to identify and classify relevant studies related to GSD project management as regards planning, controlling and monitoring. Thus, this SMS addresses seven mapping questions (MQs): publication sources, frequency of relevant literature, research approach, research method, research type, most frequently addressed SPM activities for GSD, and benefits/limits of SPM approaches in GSD.

Chapter 4 reports an overview of SCE in the context of GSD research. A mapping study has been conducted to present a wide spectrum of software estimation techniques. The selected studies were classified into nine classification criteria: publication source, publication year, research type, research approach, contribution type, software cost estimation techniques, software cost estimation activity, cost drivers and cost estimation performances for GSD projects.

Chapter 5 presents a proposed integrative framework intended to help researchers and industry practitioners identify and rank software CAs in the context of global software projects. CAs available in literature have been identified using an SLR and then analyzed and discussed to identify their impact on GSD context.

Chapter 6 reports on a survey of existing software CAs associated with the management of GSD projects in literature and their importance. The attributes selected and used in the questionnaire are extracted from the integrative framework proposed in Chapter 5 and the attributes of ISBSG dataset [19] that includes a vast number of project attributes related to the application domain, programming language used, language type, development technique, resource level, and functional size of the software produced among other.

Chapter 7 presents a taxonomy to characterize cost estimation activities for GSD context from the literature to assess whether all important aspects related to this cost estimation are reported. The taxonomy has been designed by following a taxonomy design method.

Chapter 8 concludes this thesis. This chapter contains a summary of the work presented and points its contributions to the GSD field in general and software cost estimation in particular. Finally, areas of future work are provided.

Chapter 2

Background

2.1 Introduction

The software development progressively requires globally distributed teams from around the world to meet the expectations of organizations that seek to provide customers and users with high quality software development and lower development costs [20]. This globalization increases the complexity of control and coordination in the collaborative software development effort, which can in turn negatively impact project outcomes [21]. Geographical distance, temporal differences, socio-cultural distribution, language differences, organizational boundaries, and functional boundaries inherent in GSD contexts represent significant challenges to handle and control GSD projects [22, 23]. Consequently, GSD project management is a challenging task that is a critical factor to the project success. In this respect, the software project managers of such projects must be fully aware of the challenges posed when conducting distributed activities and have the approaches to mitigate their effects [24].

This chapter is structured as follows: the recent research conducted in the field is summarized in Section 7.2, wile the research areas associated with this thesis are presented in Section 2.3. The research approaches used are described in Section 2.4. Finally, the conclusion of this chapter is presented in Section 2.5.

2.2 Related Work

Early research on software cost by GSD project managers was conducted by Keil et al. [25] in a study carried out in 2006, in which the researchers were able to structure cost drivers of distributed development and examine the significance of each of these factors as a contributor to the overall cost of a software development project. The goal of this study was to suggest ways in which COCOMO II, the most widely used software development cost estimation model, overcome difficulties and complexities confronted in distributed environment. This research was the first to deliver evidence for the lack of project factors to gain insights into comparing development costs for distributed software development projects as compared to collocated projects.

In 2008, Forbath et al. [26] published an overview to understand the successful practices used to solve cost and achieve innovation in new product development projects between leading technology firms and their global software development partners. In their study, they were able to understand the strategies and practices firms that are using to address the shift to global software engineering. The research shows that reducing costs is the main strategic goals achieved by global innovation projects. According to the cost challenges, more studies must be conducted to support innovation in global development.

Lamersdorf et al. [27] published a study in 2010 of cost factors and their estimated impact on productivity overhead model for a Spanish global software development organization. A matching between these cost drivers, their impact on the specific distributed development context was quantified. Therefore, a list was proposed to overcome the possible labor cost savings of GSD without seeing them in combination with the likely productivity downfall. In the same year, Muhairat et al. [28] tackle the same subject more extensively and classify the factors using different effort estimation methods and simulators. According to their study, effort estimation methods' accuracy is still influenced by the lacking of factors related to GSD environment. The existing effort estimation methods need improvement so that they estimate accurate effort for GSD projects. It may need to add / remove or merge the existing factors of to better match GSD requirements.

Azzeh [29] published a study in 2013, in which the author discussed three proposed factors (Global team trust, Global team composition and Culture value) that will help in managing the global software project development. These potential factors affect global project development and were identified as new factors. Recently, in 2017, Smite et al. [30] reported a number of cost drivers, when calculating the upfront business cases of offshoring large and complex software products, and especially the costs associated with mentoring. These proposed cost drivers were classified into the three following categories: The transfer of work, working on a distance, and immaturity of the offshore sites. All these studies assessed cost drivers related to distributed development project at the data capturing stage. Associating and summarizing each cost driver to a unique class give an in-depth to the software cost literature. Chapter 5 present a classification scheme of the cost drivers related to GSD projects.

2.3 Research Areas

2.3.1 Global Software Development

Today, GSD is becoming increasingly common practice in the software industry. There are a lot of software projects running in geographically distributed environments [31, 32]. There are numerous potential benefits from GSD, including reduced development costs, reduced cycle time and closer proximity to markets and customers. Many organizations have distributed software development across geographies to capitalize on global resource pools, appealing cost structures, and round-the-clock development to accomplish cycle-time acceleration and cater to local markets [10].

Over the previous decade, globally distributed software development's unique nuances, complexities, and challenges have been revealed by both research and practitioner reports. These range from technical, economic, organizational, professional, and cultural issues to those emerging from different time zones, languages, geographical locations, and national and organizational cultures [31, 33]. Besides, although it is true that a body of knowledge on GSD trend has been made over time, the art and science of organizing and managing globally distributed software development is still evolving.

The growth of GSD projects in recent years requires that many software engineers must use information and communication technologies to collaborate over geographical, temporal and socio-cultural distance, collectively termed as "global distance" [34] as illustrated in Figure 2.1. Not surprisingly, such global distances introduce major challenges in relation to team communication, coordination and control [35]. The following factors have been gathered from research literature [4, 36–39] to have an impact on globally distributed Information System (IS) development projects:

Temporal distance

- Reduced opportunities for synchronous communication.
- Typically increased coordination costs.
- Management of project artifacts may be subject to delays.

Geographical distance

- Face to face meeting difficulties.
- Lack of critical task awareness.
- Difficulties to convey vision and strategy.

Socio-cultural distance

- Cultural misunderstandings.
- Reduced cooperation arising from misunderstanding.
- Different perceptions of authority can undermine morale.
- Adaptation of managers to local regulations.
- Impact on coordination caused by inconsistent work practices.

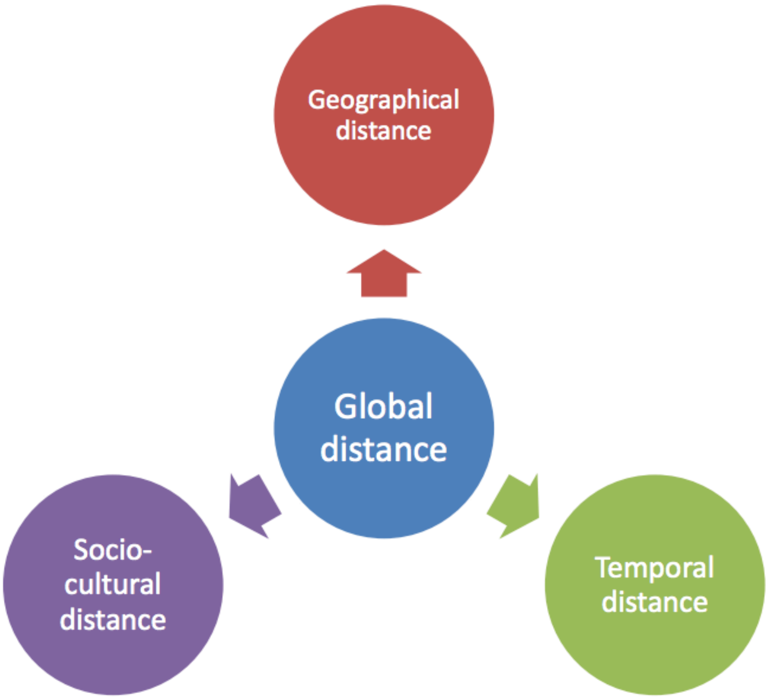

FIGURE 2.1: GSD challenges

Despite the challenges and complexities involved in organizing and managing globally software development projects, this phenomenon's pace has been remarkable [31, 40]. GSD appears to have turned into a business need for various reasons, including cost, shortage of resources, and the need to locate development closer to the customers. Indeed, the GSD trend is quick turning into a pervasive business phenomenon. In this manner, effective strategies and practices to effectively organize and manage global software development become crucial for the success of any attempt of developing software in a global context [41].

2.3.2 Software Project Management

Project Management Institute (PMI) "A Guide to the Project Management Body of Knowledge" (PMBOK) [42] defines project management as the application of knowledge, skills, tools and methods to project activities to accomplish the project's goals and objectives [43]. The effectiveness of project management is basic in guaranteeing the success of any significant activity. The project processes may generate information to enhance the management of future projects. The project management process describes the purposes and the interactions of activities organized as a series of actions conducing to a desirable end. SPM tries to gain control over five variables that affect the success of a project, including scope, time, cost, quality, and risk management. As shown in Figure 2.2, each variable represents a constraint. One constraint cannot be changed without affecting the others.

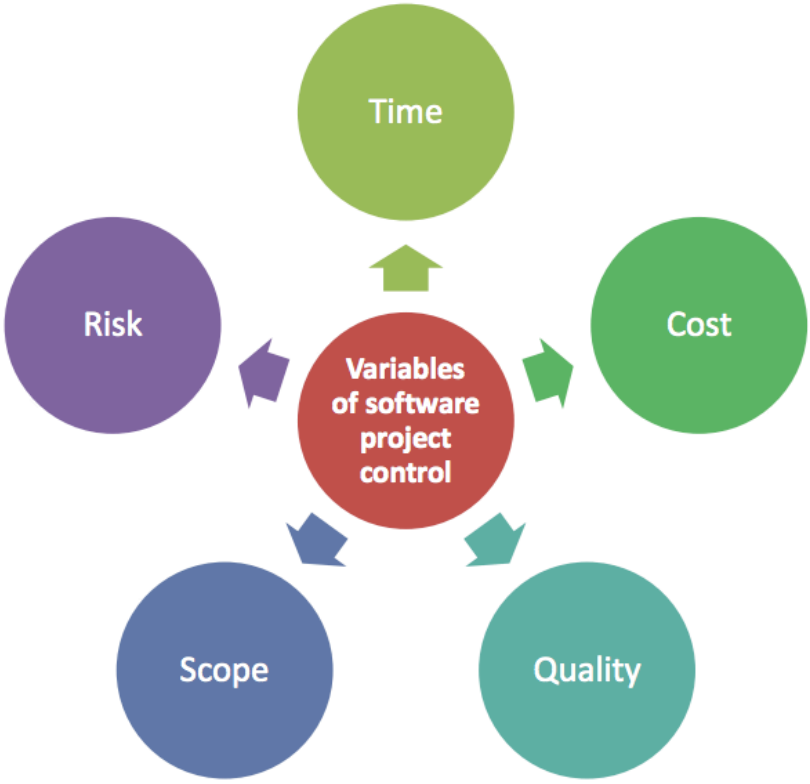

FIGURE 2.2: Variables of software project control

SPM dictate a very specific series of process groups that should be performed. These process groups provide guidance in applying appropriate project management knowledge and skills during the project. Project management processes are grouped into five categories known as project management process groups. Figure 2.3 illustrates the project Life cycle categories. These categories have clear dependencies and are usually performed in the same sequence on each project [42]. They are independent of the areas of application. Processes are often iterated before completing the project.

SPM as defined in the "Guide to the Software Engineering Body of Knowledge" (SWEBOK) [44] presents the application of management activities, planning, coordinating, measuring, monitoring, controlling, and reporting to ensure that software products and software engineering services are delivered effectively, efficiently, and also to the benefit

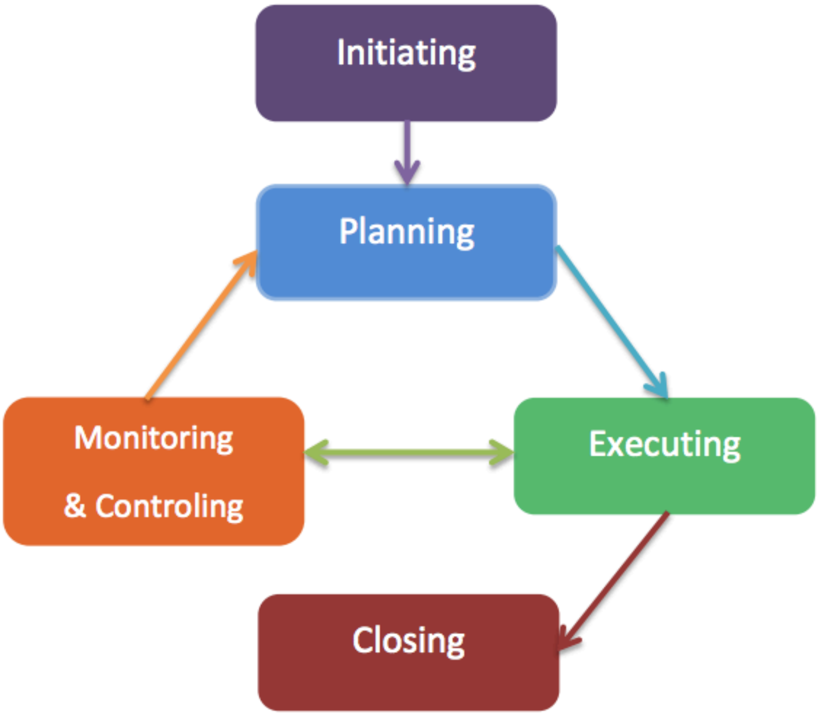

FIGURE 2.3: Project Life Cycle

of stakeholders. The international standard "ISO/IEC/IEEE 12207 Systems and Software Engineering - Software Life-cycle Process" defines eight project processes and their activities [45], as shown in Figure 2.4.

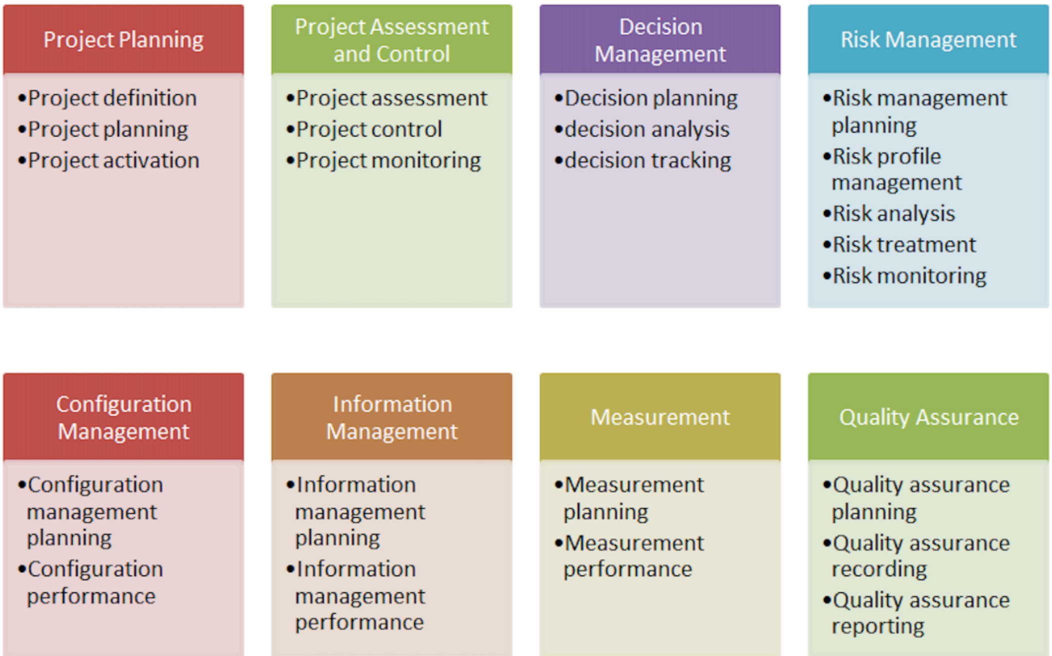

FIGURE 2.4: Activities of ISO/IEC/IEEE 12207 project processes

2.3.3 Software Cost Estimation

In recent years, software has become one of the most important component of computer system projects. The estimation of cost of software development depends on the accurate prediction of human effort, therefore, most cost estimation methods focus on this aspect and give estimates in terms of person-months [46]. In particular, for both developers and customers, accurate software cost estimation is the driving force behind generating request for proposals, scheduling, contract negotiations, monitoring and control [15]. However, software engineering cost estimation involves the determination of one or more of the following indicators:

- cost (in budget),
- effort (usually in person-months),
- project duration (in calendar time).

In order to establish an accurate cost estimate for software, a structured approach with a significant amount of work is needed. The software cost estimation can be seen as a small size project which needs to be carefully planned, managed, and followed up. The process for software cost estimation data gathered from the NASA's Handbook for Software Cost Estimation [47, 48] includes a number of iterative steps summarized in figure 2.5.

There is no simple way to make an accurate estimate of the cost required to develop a software system. Furthermore, there is a fundamental difficulty in assessing the accuracy of different approaches to cost-estimation techniques. Software cost estimates are often selective. The estimate is used to define the project budget, and the product is adjusted so that the budget figure is realized [49].

A controlled experiments with project costing where the estimated costs were not used to bias the experiment. A controlled experiment would not reveal the cost estimate to the project manager. The actual costs would then be compared with the estimated project costs. However, such an experiment is probably impossible because of the different costs involved and the number of variables that cannot be controlled.

Table 2.1 describes the techniques [49, 50] used to make software effort and cost estimates. All of these techniques depend on experience-based judgements by project managers who use their insight into previous projects to land at an estimate of the resources required for the project.

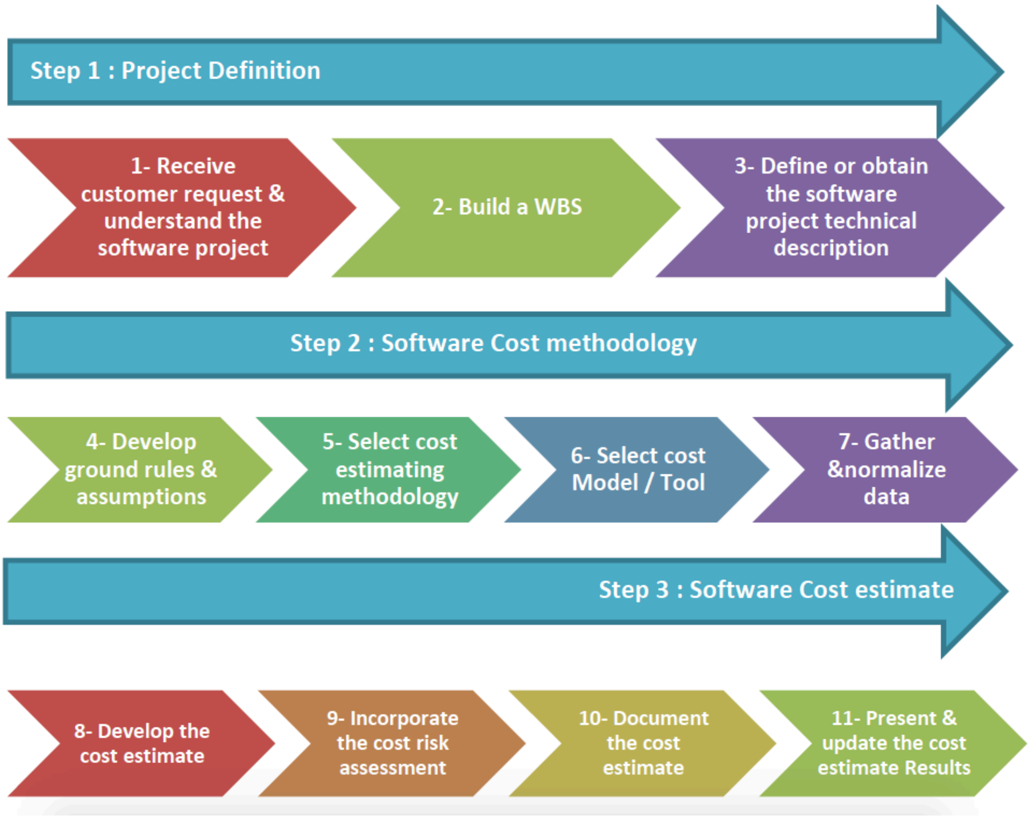

FIGURE 2.5: Overview of Software Cost Estimation steps

2.4 Research Approaches

2.4.1 Systematic Reviews

The main goals of a systematic review is to ensure that the review is methodical, repeatable, and thorough. It also attempts to minimise the level of bias that can be prevalent in traditional literature reviews [51]. As the number of published literature reporting empirical studies grew in Software Engineering (SE) topic, systematic review has been gaining significant attention from researchers [52]. Both systematic literature studies, SMSs and SLRs [53], have been emerged as popular methods of Evidence-Based Software Engineering (EBSE). SMSs and SLRs share some similarities in regard to the search strategy and the study selection. However, They present contrasts as for the RQs, search process, search strategy requirements, quality evaluation, and results. The following subsections describe the two types of systematic studies.

Systematic Mapping Study

An SMS is a literature study that is focused on selecting and synthesising all high quality research related to a particular topic and providing an exhaustive summary of

Table 2.1: Software Cost Estimation Techniques

Techniques	Description
Algorithmic cost modelling	A model is developed to provide mathematical equations to perform the estimation. These equations are based on research and historical cost information that relates cost drivers. An estimate is made of software metric and the model predicts the effort required.
Expert judgement	Software cost estimation experts are consulted. They each use their experience and estimate the project cost. These estimates are analyzed and discussed. The estimation process iterates until an concurred estimate is reached.
Estimation by analogy	This technique means comparing project to similar previously completed project. The cost of a new project is estimated by analogy with data extracted from the completed projects.
Parkinson's Law	Parkinson's Law states that 'work expands to fill the available volume'. The cost is controlled by available resources instead of by objective assessment.
Pricing to win	The estimated cost depends on the price that is necessary to win the software project contract and not on the software functionality.

current literature that is relevant to specific mapping questions, using explicit methods to identify what can reliably be said. The principal goal of an SMS is to provide a formal means of synthesising the information available from convenient primary studies relevant to a set of mapping questions [54, 55]. This method is able to cover three main phases: planning, conducting and reporting [56]. The objective of these phases is to identify, evaluate and interpret all available research relevant to a particular topic based on the strength of their evidence, draw conclusions and finally provide recommendations. The phases of the SMS process are presented in Figure 2.6 [57, 58].

SMSs are designed to identify, evaluate and interpret all available evidence related to a particular research question [59]. They are generally used to examine the extent to which empirical evidence supports/contradicts theoretical hypotheses, or even to assist the generation of new hypotheses. Kitchenham et al. [59] stresses the importance of identifying and reporting all available research including evidence that does not support the preferred research hypothesis.

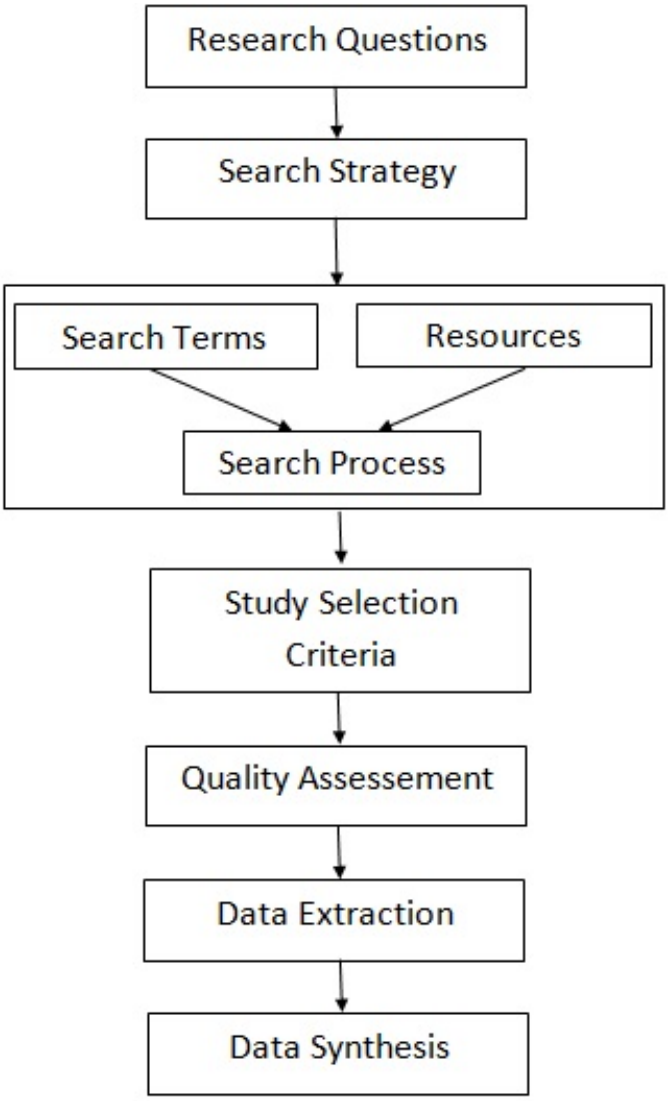

FIGURE 2.6: Stages of systematic mapping process

Systematic Literature Review

SLR is a means of assessing and interpreting all existing research relevant to a specific research question, topic area, or phenomenon of interest. The aim of an SLR is to support the development of evidence-based guidelines for practitioners. The end point of EBSE is for practitioners to use the guidelines to provide appropriate software engineering solutions in a specific context, especially the GSD one [16]. The main reasons for undertaking a systematic literature review are:

- Summarise the existing evidence concerning a treatment or technology and setting the review questions;

- Identify relevant work and any gaps in current research in order to suggest areas for further investigation;
- Provide a framework/background in order to appropriately position new research activities.

2.4.2 Framework

A framework is a well suitable research methodology for many kinds of software engineering research [60, 61]. A framework is an empirical inquiry that provides a conceptual foundation by which a researcher and practitioner can understand, conduct and assess action research, these research approach can also enrich the process of learning beyond the traditional academic boundary and contribute to theory and practice [62].

The theoretical framework, as distinct from a theory, represents the choice of paradigm that sets down the intent, motivation and expectations for the research. Kitchenham and Pfleeger [63] define a framework as a comprehensive system for:

- how to validate a measure;
- how to assess the validation work of others;
- when it is appropriate to apply a measure in a given situation.

2.4.3 Surveys

Survey is a common research approach in academic research and otherwise. Pfleeger and Kitchenham [64, 65] define a survey as a comprehensive system for collecting information to describe, compare or explain knowledge, attitudes and behavior. Thus, In order to be successful, survey research process is divided into seven steps:

- Step 1: Setting and planning
- Step 2: Designing
- Step 3: Collecting data
- Step 4: Validating data
- Step 5: Administering and managing data
- Step 6: Analyzing data

- Step 7: Reporting results

Survey is an appropriate means of gathering information from a pre-determined population “population of interest” (POI) [66, 67]. The results from the survey are then analyzed to derive descriptive and explanatory conclusions. Then, these results are generalized to the population from which the sample was taken [68]. Figure 2.7 provides the advantages and disadvantages of several survey’s methods.

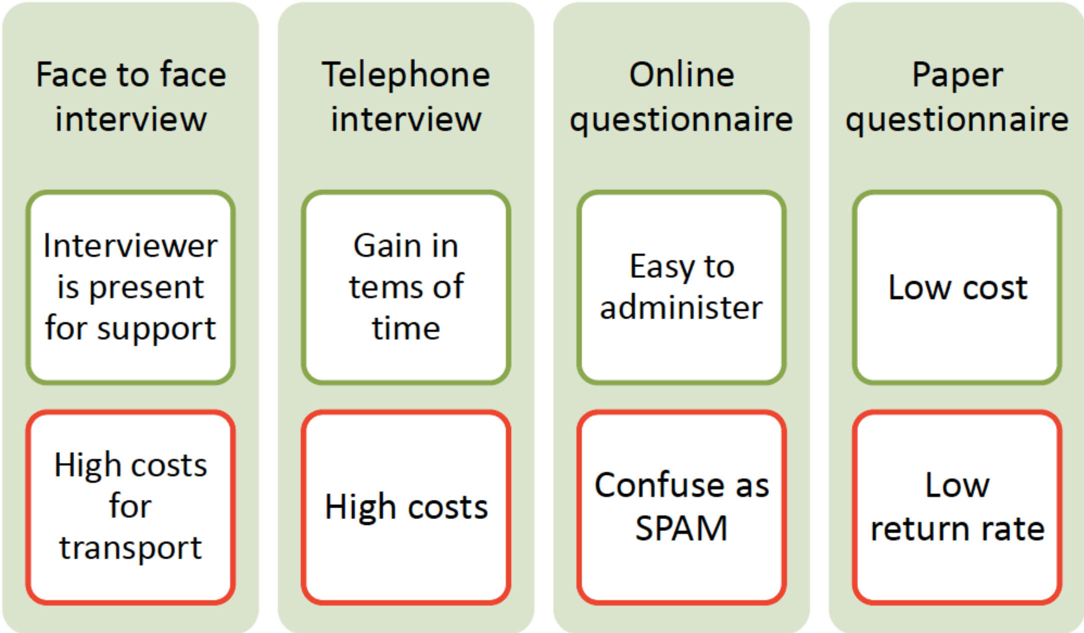

FIGURE 2.7: Survey methods

There are several types of surveys. Surveys can be either supervised or not, depending on the objectives and the resources available. If supervised, we can assign one survey researcher to each respondent, to ensure that the respondent understands each question and provides an answer. Surveys provide four key characteristics: (1) response rate, (2) timeliness, (3) data quality, and (4) cost.

2.5 Conclusion

GSD projects suffer greatly from several issues like communication, coordination and trust. Different challenges are highlighted by different researchers and some of researcher give solution or to avoid these challenges face by vendor and clients to GSD projects in off-shoring outsourcing, onshoring outsourcing and near-shoring [32, 69, 70]. In this regard, GSD Project managers are not only in need of approaches that support the project processes, but also in need of approaches developed specifically for stakeholders when it perform different geographical location, cultural differences and time zones.

In this Chapter, we gave an insight on research areas related to this study, described the research approaches used and reported on the recent research carried out in this field. In the next chapter, we give an overview of SPM approaches used in GSD context.

Chapter 3

Overview of SPM approaches used in GSD

3.1 Introduction

Global software development (GSD) refers to software development that is dispersed over at least two locations that are separated by national or continental borders [71]. Interest in GSD is rapidly growing as the software industry is experiencing increasing commercial globalization. In GSD, stakeholders from different national and organizational cultures and time zones are involved in software development. In this scenario, tasks at various stages of the software lifecycle may be separated for development at different geographic locations, and then coordinated using information and communication technologies [72]. While increasing the scope of organizational operations and opening up opportunities for a broader skill and product knowledge base, GSD also poses real challenges related to project diversity and complexity. As such, attention must be given to how to enable the successful management of software projects. According to the European Space Agency (ESA) guide to Software Project Management (SPM) [73], SPM is a "process of planning, organizing, monitoring, controlling, and leading a software project". To be successful, software projects must effectively coordinate numerous activities by multiple organizational actors or units. SPM can also be defined as a system of procedures, practices, and technologies that address the management and measurement of software engineering [74]. SPM for GSD, in turn, involves the participation of different globally distributed managerial and technological resources to produce software of the highest possible quality with the minimum cost and development time [75, 76].

Organizations are constantly seeking ways to obtain a larger pool of skilled professionals, optimize costs, and reduce delivery times. as such, GSD projects have become a widespread reality [21]. Working in a global context has advantages as well as drawbacks [77]. On one hand, time-zone effectiveness is gained and cost is reduced in various countries. On the other hand, working on a globally distributed project [78] means increased operating costs with respect to planning and managing people due to language and cultural barriers. These differences can also create jbad feelings, as more highly trained and paid engineers (who are concerned about their own job security) are asked to train their much less capable and costly counterparts [79].

Identifying the critical aspects that contribute to the success of a software project is a critical step for successfully solving the problems associated with GSD. Some SPM approaches (methods, models, frameworks, and processes) for GSD have been identified for dealing with challenges related to communication, coordination, collaboration, and performance in GSD [80]. The growing globalization of SPM has attracted a great deal of attention, and has led to the demand for specific techniques for planning, communication, coordination, and control in the management of these projects. For a global software project is to succeed, the challenges related to SPM for GSD must be identified and insights gained.

The main objective of this chapter was to conduct an exhaustive review and synthesize the body of current research addressing GSD project management with respect to its planning, control, and monitoring practices. The classification of related activities should be based on the main SPM bodies of knowledge and standards. We conducted a systematic mapping study (SMS) to facilitate our summary of the approaches proposed for the management of distributed software projects and to address SPM deficiencies with respect to GSD [52]. Via the SMS process, we identified the most frequently addressed SPM approaches in the GSD context. The SMS study results also revealed whether the authors of selected papers based their solutions on criteria for project management, decision models, or computational techniques. Our intention in this project was to help identify suitable results and gain insight into topics to then provide an overview and a set of leading recommendations. The articles identified in this study generally present a single SPM topic related to the GSD problem, or covers two to three types of the areas in question. Moreover, the papers we reviewed address how to deal with problems associated with SPM for GSD by introducing techniques and methods, and sometimes by describing these methods and their impact on the project results.

During the life cycle of a globally distributed project, project management and control activities must be given special consideration by companies to ensure effective product development. to achieve the fixed goals of a project, project management must involve multiple activities, including the planning, scheduling, organizing, controlling, and managing of tasks and resources. Currently, based on the topics addressed in the project management problem area of distributed software development, the most important activities identified in GSD projects are planning, controlling, and monitoring, i.e., detailed project planning and strict control and monitoring during the project [81]. Research published in the literature highlight this finding, as more than 50% of researchers have investigated these activities and mention them as being most important for analysis and discussion [82]. This observation prompted our decision to focus on these three activities.

This chapter is structured as follows: we present related work in Sect. 3.2, and identify the main SPM bodies of knowledge and standards in Sect. 3.3. We present our study research method in Sect. 3.4 and report the results we obtained from the SMS in Sect. 3.5. We discuss our main findings in Sect. 3.6, as well as the resulting implications for researchers and practitioners. We identify threats to validity in Sect. 3.7 and present our conclusions and thoughts on future work in Sect. 3.8.

3.2 Related work

Although the topic of SPM for GSD has been studied and discussed for many years, we found few literature reviews or surveys of SPM approaches for GSD. In this section we summarize the most relevant work to date.

Hossain et al. [83] studied the use of SCRUM in GSD, having found agile practices to be extremely popular in the GSD domain. The main conclusion of the authors was that it is difficult to provide solutions for GSD challenges because of the different types of development distributionsacross projects. The systematic literature review (SLR) by Schneider et al. [84] focused on identiyiing challenges in global software projects and popular research areas. For researchers, in particular, the SLR provides a model on which to build and provides insight regarding which process areas are well researched. For industry practitioners, the SLR can serve as a reference framework for avaluating and improving the SPM development environment by identifying solutions in a structured manner [85].

Niazi et al. [79, 86] conducted an empirical study to identify SPM barriers in GSD that may undermine software project implementation initiatives. The authors developed a model to measure organizations' project management readiness for GSD activities, and provided a body of knowledge that can help practitioners design and implement successful SPM initiatives. To avoid the risks associated with software process improvements, SLRs also suggest key factors regarding which management areas should receive more attention. Vizcaíno et al. [87] conducted a survey of the opinions of 21 experts in SPM for GSD. These authors also analyzed the relevant GSD success factors reported in literature that are mainly related to SPM.

Some SLRs have focused on the challenges and improvements associated with distributed projects. Jiménez et al. [21] reviewed the available literature to identify the solutions and improvements proposed up to 2009. Their paper also identifies the interest in modeling software development, and the relative benefits of approaches that dress improving productivity, quality, and the level of understanding between the team members involved in the development process. Darja et al. [88] reported their empirical findings regarding the global software engineering (GSE) related literature in terms of useful practices or techniques. Seven practices were highlighted and discussed as prerequisites for success.

In an SLR published in 2010, da Silva et al. [89] presented the best practices used in the management of distributed software development projects and their associated challenges. This SLR provided a good overview of the SPM landscape and identified the need to devise experiments to quantify evidence regarding the effect of using best practices and models.

In summary, after considering the research and publications on SPM for GSD to date, our work serves as a starting point for determining the current SPM approaches used in the GSD context. The qualitative insights provided by the above studies are particularly relevant to our research and must be taken into account when generalizing the findings of this paper. However, there has been scant work to date to comprehensively synthesize and summarize the state-of-the-art of SPM for GSD. None of the aforementioned SLRs provide any classification of activities identified in the Project Management Body of Knowledge (PMBOK) [90], nor any synthesis of the benefits and limitations of the SPM for GSD approaches selected. On one hand, studies such as [89] focused on approaches and practices, but do not go into the issue in any depth or classify them according to relevant standards, such as PMBOK. On the other hand, a few studies have focused on SPM challenges but their finding are not sufficiently recent and their coverage of this issue is limited. To date, literature reviews have focused on specific SPM areas, such as agile management or estimation [27, 91], but there has been no systematic review that provides a more complete coverage of the main SPM areas. Given these observations, we realized the importance of using the SMS framework as a means for gaining new insights into the specification and the classification of SPM approaches for GSD. Therefore, using the PMBOK as a framework, our work consisted of collecting, synthesizing, and classifying the current and most relevant knowledge regarding SPM approaches for the GSD context.

3.3 SPM bodies of knowledge and standards

In this section, we present our analysis of the current situation in SPM, based on the most important project management bodies of knowledge (BOKs) and standards. BOKs, standards, and related assessments can be viewed as essential building blocks in the formation and recognition of a distinct SPM profession. Various organizations have worked diligently to identify related software engineering knowledge. Many BOKs and standards that include SPM content have been established to assist managers to successfully undertake SPM activities. A BOK is defined as a complete set of concepts, terms and activities that comprise a professional domain [92], whereas standards are the result of a consensus that has been formally approved by a recognized body with the aim of achieving the optimum degree of order in a given context [93]. Overviews of the BOK and standards that contribute to the development of a better SPM structure are presented in the subsections below.

3.3.1 Bodies of knowledge

- The PMBOK [90] contains the sum of knowledge in the management profession, and is divided into five basic process groups: initiating, planning, executing, monitoring/controlling and closing. Each process group is divided up into ten management knowledge areas: integration management, scope management, time management, cost management, quality management, human resource management, communications management, risk management, procurement management, and stakeholder management .
- The Software Engineering Body Of Knowledge (SWEBOK) [94] describes knowledge in the field of software engineering, and includes ten Knowledge Areas (KAs), including engineering management, engineering process, configuration management, and quality management. The software engineering management KA contains five subareas: initiation, planning, enactment, evaluation, and closure.

3.3.2 Standards

- The IEEE Std 12207-2008 [95] is a guideline that can be used to define, control, and improve software life-cycle processes. This standard is applied to the acquisition of systems, software products and services, to the supply, development, operation, maintenance, and disposal of software products and to the software portion of a system, whether it is performed within or outside the organization.

- IEEE Std 15288-2008 [96] is a process standard that is intended to help organizations and projects establish an appropriate environment for the desired processes. This standard defines a set of processes and associated terminology for the full project life cycle, including conception, development, production, utilization, support, and retirement. It also supports the definition, control, assessment, and improvement of these processes.
- ISO/IEC/IEEE 16326 [97] is a guideline for project management plans that cover software projects, and software-intensive system projects. This guideline cancels out and replaces ISO/IEC TR 16326.
- ESA PSS-05-08 [73] is an ESA guideline describing the software engineering standards to be applied in the implementation of all ESA's deliverable software.
- ISO 21500:2012 [98] is a project management standard that provides high-level descriptions of the concepts and processes that constitute good practices in project management, which can be used by any type of organization, whether public, private, or community-based, and for any type of project, irrespective of its complexity, size, or duration.

The scope of this chapter with respect to the software project issues of planning, monitoring, and controlling processes point to the PMBOK as being the most appropriate body of knowledge for area classification. In this SMS, we used the PMBOK process to group reference categories for classifying software project management proposals. Although other options are available, the PMBOK provides a standard reference with which to classify project management processes according to the project lifecycle, and is widely recognized by industry and academia. PMBOK recognizes processes that fall into five basic process groups, including the three we selected: planning, monitoring, and controlling. The objective of our study, therefore, is to identify SPM approaches in the literature that are related to the GSD field and classify them with reference to the PMBOK.

3.4 Research methodology

SMSs are designed to provide a formal means of synthesizing the information available from convenient primary studies relevant to a set of mapping questions [99].

3.4.1 Mapping questions

There are many ways to organize and manage distributed development. GSD can be conducted via different scenarios and be implemented in different organizational forms. The aim of implementing GSE scenarios (with respect to cost, resources, communication, quality, etc.) is to realize SPM success [100].

In this study, our goal is to gain insight into the existing SPM approaches for GSD. In particular, in this chapter, we focus on mapping questions related to developing and evaluating a classification scheme for GSE-related studies. To validate and improve the scheme, we conducted a review, for which we selected studies reported at international conferences and in professional journals, since these particular conferences and journals focus explicitly on publishing high-quality work. Furthermore, we sought to understand existing research directions within the field of GSE and the research specific to this topic [88]. The latter is particularly important since it provides evidence about what we actually know. Due to the differing strengths of the studies, findings may vary with respect to the research methods, types, and approaches. The strength of the empirical evidence in a field provides important information for making decisions about future research and how to practice globally distributed development. Thus, this SMS addresses seven mapping questions (MQs), which are presented in Table 4.1, along with their principal motivations. These questions allow for the categorization of current research into SPM techniques for GSD and the identification of future areas of research in the field, which we used as a basis for defining the search strategy and paper selection criteria. Therefore, we emphasize the importance of describing the methods used to gather and analyze empirical data (e.g., survey, case study, experiment, or other).

3.4.2 Search strategy and paper selection criteria

The extracted papers were identified using the specified search terms, and the searches have been performed in the following sources in January 2016:

- IEEE Xplore digital library [1]
- ACM digital library [2]
- ScienceDirect [3]
- SpringerLink [4]

[1] http://ieeexplore.ieee.org
[2] http://dl.acm.org
[3] http://sciencedirect.com
[4] http://link.springer.com

Table 3.1: Mapping questions

ID	Mapping question	Motivation
MQ1	Which publication channels are the main targets of SPM approaches for GSD?	To identify where studies on SPM approaches for GSD research can be found in addition to the suitable targets for publications of future studies.
MQ2	How has the frequency of SPM approaches for GSD publications changed over time?	To identify the publication trends of SPM approaches for GSD research over time.
MQ3	Which approaches have been used for SPM in the GSD context?	To discover Software Engineering techniques, methods and models used in SPM for GSD.
MQ4	What research methods are used in the selected papers?	To identify existing research methods reported in the existing literature.
MQ5	Into which research type are SPM approaches for GSD classified?	To explore the different types of research reported in literature concerning SPM techniques for GSD.
MQ6	Which SPM activities for GSD as regards planning, monitoring and controlling were most frequently addressed?	To discover the most frequently existing SPM for GSD activities.
MQ7	What are the benefits and limits of SPM approaches in GSD?	To provide a detailed analysis that identifies the benefits and limits reported in the existing literature.

- Google Scholar [5]

These sources were selected on the basis of systematic reviews in the same field with a similar scope. The researchers involved in the studies [21, 88, 101] used IEEE, ACM and ScienceDirect libraries.To include more results, we also used two additional search sources (SpringerLink and Google Scholar), from which we identified a set of professional software engineering journals and events (such as ICGSE, ICSE, CSCW, CCECE, APSEC and ESEC).

To answer the MQs, we searched for research papers with a background and main focus on GSD and SPM. Although the search terms selected for extracting the relevant papers do not cover all possible research methods, we consider them to be sufficient in their coverage of in-depth studies related to our research scope, since we were seeking papers with depths characteristic of those identified in an SMS.

[5] http://scholar.google.com

We grouped search terms with similar meanings and obtained combined terms using the OR logical operator between search terms in the same group. To perform automatic searches in the selected digital libraries, we formulated a search string using the AND logical operator between combined terms of different groups, as follows:

(Software **OR** system* **OR** application*) **AND** (project* **OR** process* **OR** product*) **AND** (manag* **OR** improv* **OR** assess* **OR** develop* **OR** monitor* **OR** plan* **OR** control* **OR** coordinat* **OR** perform*)**AND** (technique* **OR** method* **OR** need* **OR** approach* **OR** factor* **OR** model* **OR** strateg* **OR** best practices **OR** measur*) **AND** (Global development **OR** Global engineering **OR** distributed development **OR** outsourc* **OR** Offshor* **OR** Dispersed development).

This search string was inspired by those used in similar research [89, 102] and also from author suggestions. We applied this search string to the titles, abstracts, and keywords of papers to reduce the search results [57]. Each paper was retrieved by the first author and specific information about each relevant paper was extracted and entered in an MS Excel file, as shown in Fig. 3.1.

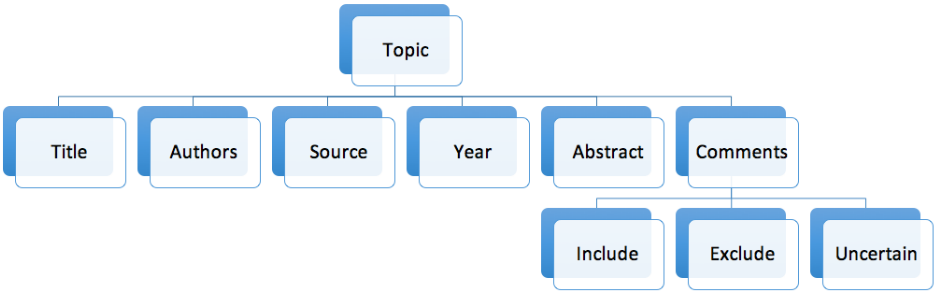

Figure 3.1: Fields on the selection sheet

Two authors were each responsible for retrieving papers by considering each paper's title and abstract. When there was disagreement, the full text was accessed to reach an agreement. The paper was then included, excluded, or classified as uncertain. Two other researchers were asked to review the selected papers on the basis of their titles and abstracts. Paper were included if both researchers agreed that the study was relevant and was excluded if both researchers agreed that the paper was irrelevant. Papers that were judged differently were discussed until an agreement was reached. The Kappa coefficient for this selection process was 0.9, which indicates almost perfect agreement between the two assessments [103]. The final selection was reviewed by the remaining authors involved in this study.

The first step after the application of the search string was to eliminate duplicate titles and papers that were clearly outside the study scope. Potential primary studies were then selected using the Inclusion / exclusion criteria. To focus on studies that had presented management mechanisms, paper selection was accomplished without considering the development aspect.

We defined the following inclusion and exclusion criteria based on established SMS guidelines [52]. Note that the inclusion criteria are linked by an "and" to join the main criteria and thereby ensure the pertinence of the selection, whereas the exclusion criteria are linked by an "or" to indicate that compliance with only one criterion is sufficient for the paper to be excluded.

Inclusion criteria

- IC1: Papers related to the SPM aspect of GSD projects.
- IC2: studies which tackle SPM approaches related to the following PMBOK process groups: "Planning" and "Monitoring and Controlling".

Exclusion criteria

- EC1: Papers that are not published in journals, conferences or workshops.
- EC2: Papers that are a workshop summary.
- EC3: Papers that are not in English.

3.4.3 Quality assessment

Quality assessment in a systematic review is a major focus that increases its depth. To enhance our study results, we designed a questionnaire to assess the quality of candidate papers. We used a scoring technique based on those used in previous studies [104, 105]. The scoring plan, as shown in Table A.6 in Appendix A (columns a, b, c, and d), is as follows:

(a) The paper has been published in a recognised and stable journal or conference. This question was rated by considering the computer science conference rankings in Computing Research and Education (CORE) 2013 Conference Rankings [6], and the 2013 Journal Citation Reports [7] (JCR) lists. The possible answers to this question were:

[6] http://www.core.edu.au/coreportal
[7] http://webofknowledge.com/JCR

- For conferences and workshops:
  - "Very relevant (+2)" if it is ranked CORE A*,
  - "Relevant (+1.5)" if it is ranked CORE A,
  - "Not so relevant (+1)" if it is ranked CORE B,
  - "Not relevant (+0.5)" if it is ranked CORE C,
  - "No ranking (+0)" if it is not in CORE ranking.
- For journals:
  - "Very relevant (+2)" if it is ranked Q1,
  - "Relevant (+1,5)" if it is ranked Q2,
  - "Not so relevant (+1)" if it is ranked Q3,
  - "Not relevant (+0.5)" if it is ranked Q4,
  - "No ranking (+0)" if it is not in the JCR ranking.

(b) The main focus of the paper is the SPM approach used in the GSD context. The main interest of the paper was to discuss SPM approaches in a GSD context. Yes (+1); Partially (+0.5); No (+0)

(c) The paper presents and/or explicitly assesses an approach as a solution to deal with GSD project management challenges. The study obtains the full score if it presents a new or assesses an existing approach. Yes (+1); No (+0)

(d) The study is empirical and presents relevant data for our SMS. The empirical results involves the use of SPM approaches for GSD. Yes (+1); No (+0)

3.4.4 Data extraction strategy

The data extraction strategy was based on presenting the set of possible answers to the MQs. A spreadsheet was used to extract data concerning each article which is presented in Table 3.2. The strategy is explained below:

TABLE 3.2: Paper data extraction form

Paper Id	Authors	Title	Publication source	Year	Approach	Research method
**	**	**	MQ1	MQ2	MQ3	MQ4

Research type	Activity	Benefit/ Limit
MQ5	MQ6	MQ7

MQ1: Publication source and channel were identified for each paper.

MQ2: Articles were classified per year in order to infer the publication trend.

MQ3: Approaches were classified using [1] as a basis:

- Data mining techniques: A number of complex algorithms and techniques are used to help software companies establish data patterns and trends. Each of these techniques analyses data in different ways.
- Methods: A means of procedure and series of steps are taken to best match the characteristics and contexts of SPM for GSD.
- Models: A description of a system or process that allows the inspection of SPM activities for GSD.
- Process: Series of actions, techniques, or functions leading to SPM result and performing operations on data.
- Framework: A real or conceptual structure intended to serve as a support or guide for the building of SPM for GSD.
- Other: e.g. Tool-based techniques and any approach not listed above.

MQ4: The research methods used in SPM for GSD can be classified as in [106]:

- Case study: An empirical inquiry that investigates an SPM phenomenon within a real GSD context.
- Survey: A method for collecting quantitative information concerning an SPM technique for GSD.
- Experiment: An empirical method applied under controlled conditions, using subjects to evaluate an SPM approach for GSD.
- Other: e.g. Theory and any research method not listed above.

MQ5: Every research type can be classified into the following categories [52, 107]:

- Evaluation research: Existing or new approaches implemented in practice. An evaluation or a validation of each technique is conducted including the comparative studies and analysis of SPM approaches for GSD.

- Solution proposal: When a solution is proposed, it may be a new SPM approach for GSD or a significant extension of an existing approach.
- Experience: Papers expressing the author's personal experience and explaining what has been done and how this was realised in practice.
- Other: e.g. Theoretical papers or opinion paper.

MQ6: SPM is the part of project management during which software projects are planned, implemented, monitored, and controlled. After identifying the scope of the project, estimating the work involved, and creating a project schedule, a project plan is developed to describe the tasks that will lead to completion. The objective of project monitoring and control is to keep the team and managers updated on the project's progress. Every time a change is required, change control is used to keep track of the product updates. To ensure the success of the project in a GSD setting, particular care must be taken in these more challenging areas. We chose the PMBOK since we believed it would help us to identify a reference model with which to present our work.

The main SPM activities for GSD we selected for the study are the PMBOK areas [108] related to the process groups of planning, monitoring, and controlling. We excluded the project risk management knowledge area since there is a wide body of work in the literature concerning risk management and GSD, which has been reviewed in a number of studies and SLRs, such as [109] and [110]. The study areas we adopted are shown below:

- Project Integration Management, whose purpose is to include the processes needed to identify and coordinate the various project management activities:
  - Develop project management plan
  - Monitor and control project work
- Project Scope Management, whose purpose is to include the processes needed to ensure the inclusion of all the work required in the project. In order to complete the project successfully, the steps followed are:
  - Plan and scope management
  - Validate and control scope
- Project Time Management, whose purpose is to include the processes required to accomplish the timely completion of the project:
  - Define activities and plan schedule management

  - Control schedule

- Project Cost Management, whose purpose is to include the processes involved in planning, estimating, budgeting, and controlling the costs of the project. It can be applied to:
  - Plan, estimate and determine costs
  - Control costs
- Project Quality Management, whose purpose is to include the organisation activities that will determine quality policies, objectives, and responsibilities in the project:
  - Plan quality management
  - Control quality
- Project Human Resource Management, whose purpose is to include the processes that are used to organise and manage the project team by planning human resource management:
  - Plan human resource management
- Project Communication Management, whose purpose is to explore the processes required to ensure the timely and appropriate generation, collection and distribution of project information and to reduce gaps between the exchange of people:
  - Plan communications management
  - Control communications

MQ7: A detailed analysis of the qualitative findings of the papers included in the review is provided by identifying the benefits and limitations of the approaches used in SPM for GSD, as described by the authors.

3.5 Results

In this section, we present and discuss the findings of this review. First, we present an overview of the selected studies and then we report the review findings for the MQs listed in Table 4.1.

3.5.1 Overview of selected studies

The search string that was applied to the various digital libraries and search engines returned a high number of results (54,233). These papers were published during the time period 1998-2015. 84 of these papers were eventually selected and identified as being relevant to our subject.

Fig. 3.2 shows an overview of the search process and presents the number of studies remaining after each step of the selection process. In total, after the application of the inclusion criteria, 239 papers concerning SPM for GSD were identified and placed in the Excel file. When the same paper appeared in more than one source, it was considered only once according to the order of sources. The exclusion criteria were then used to exclude another 155 studies. The final result is therefore 84 selected studies. Table A.1 in Appendix A shows the results per source after the inclusion and exclusion process.

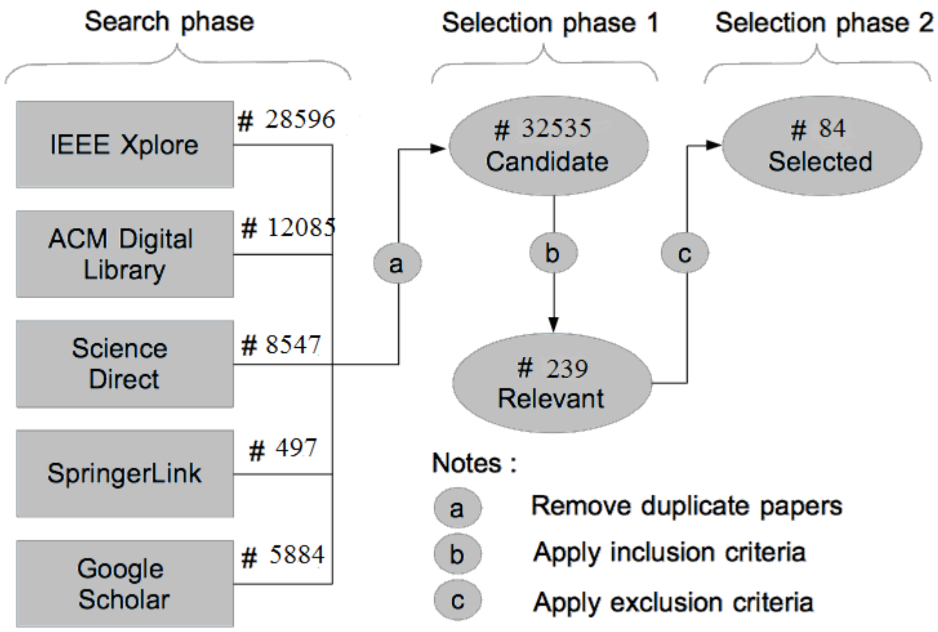

FIGURE 3.2: Search and selection process

Table A.6 in Appendix A presents the list of the selected papers with detailed information on their classification results and quality assessment. Note that 70% of the papers have a score higher than or equal to the average score of 2.5 points.

All systematic reviews emphasise that it is essential to assess the quality of the studies selected. This quality assessment is a major step as regards obtaining a general view of the paper's implication in the subject [52]. Table A.2 in Appendix A shows a five-level quality classification. Although 18 of the papers selected have a low quality level, they contain some useful information, particularly as regards software projects and that which has an important value in characterising the link between SPM and GSD. The quality classification scheme of the selected studies shows that 97,6% of the relevant papers have a score strictly higher than 1. Detailed scores for each of the studies selected are shown in Table A.6 of Appendix A.

3.5.2 Publication channels (MQ1)

As shown in Table A.3 of Appendix A, 63% of the selected papers are presented in conferences, which indicates that this is the most frequent source. Table A.4 in Appendix A presents the journals and conferences in which the papers selected for this SMS were published. With regard to journals, The IEEE Software journal and the Information and Software Technology journal are the most recurring publication sources for the SPM for GSD topic, while in the case of conferences, the International Conference on Global Software Engineering (ICGSE) is the most frequent source to attract publications concerning the topic.

3.5.3 Publication trend (MQ2)

According to Fig. 4.1, interest in SPM for GSD began in 1998. Smite et al. [88] mentioned that globalisation is considered to be a recent field. In fact, the most relevant studies about the subject have been conducted since 2000. The interest in the topic increased to reach its peak in 2006, the year of the first edition of the ICGSE conference. Interest in the subject began to stabilize in 2009.

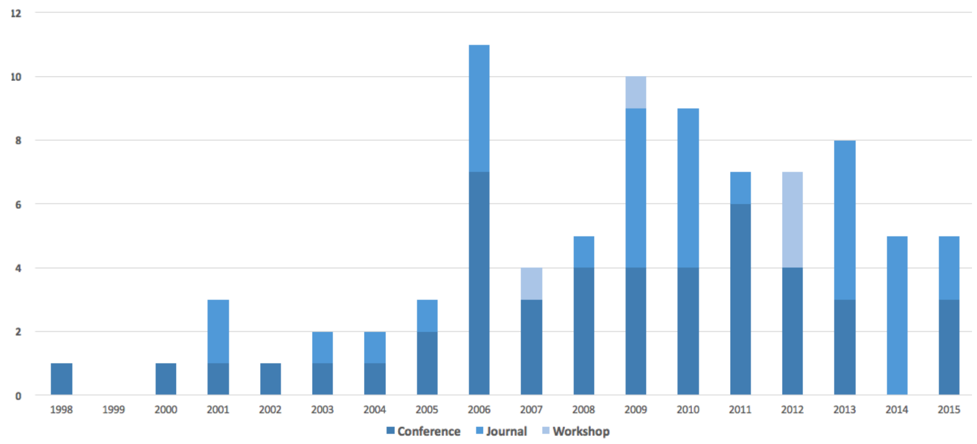

Figure 3.3: Publication per year

3.5.4 SPM approaches for GSD (MQ3)

Fig. 4.2 is a representation of the SPM approaches for GSD categories based on the papers selected. Thirty nine percent of the selected studies fall within the ”method” category. A closer look at this category shows that the majority of these methods are agile methods (61%) that rely on communication. There are also several data mining techniques (14%), mainly those that help to solve the challenges of software cost estimation and effort estimation [25, 111, 112], which are crucial activities in the software development life cycle. These techniques also help to reduce costs and improve both productivity and quality in GSD (e.g. Genetic Algorithms (GA) and Case-Based Reasoning (CBR)).

We also identified models [113–115] for GSD representation, which standardize and systematize the requirement specifications of the interaction between work teams that are physically distant from each other. These models support project planning and process improvement in GSD. The model approach, which represents almost 19% of the total categories, takes into consideration the system dynamics and continuous factors and their interactions, including communication, coordination, cultural issues, the learning curve, changing staff levels, and dynamically varying productivity [114]. Indeed, the most frequently used models are estimation models (26,6%).

About 13% of the categories were processes, principally the Delphi process. Finally, frameworks comprise nearly 7% of the categories, particularly the resource-based view (RBV) and dynamic capabilities frameworks.

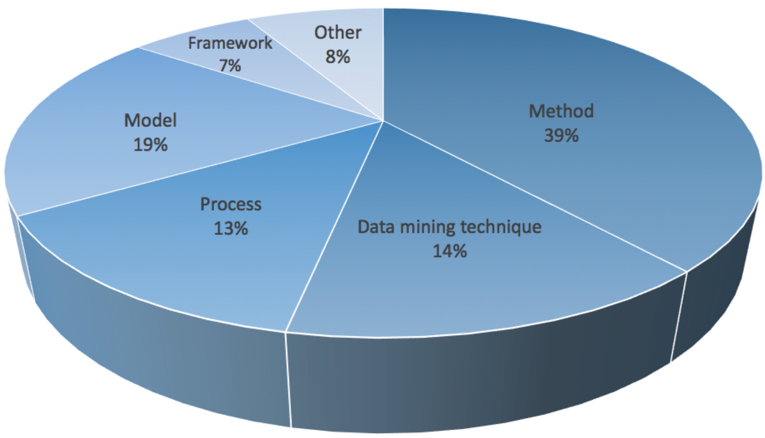

FIGURE 3.4: SPM for GSD approaches

3.5.5 SPM approaches for GSD research methods (MQ4)

Almost 94% of the selected studies concerning SPM approaches for GSD belong to one of the three main research methods: case study (48 papers), survey (10 papers) and experiments (6 papers), as shown in Fig. 3.5. A type named "other" is used to include the remaining research methods such as theoretical papers.

Seventy percent of the selected papers include case studies, which demonstrate the feasibility of the approach. Each case is conducted to evaluate a concept. It starts by evaluating goals and methods, then describing the project and finally executing the project in order to show results and discuss implications [116]. Fifteen percent of our selected papers include surveys.

A procedure considering costs and dependencies between projects must be elaborated [117] if the planning, the scheduling of distributed tasks and the application of corrective measures and notifications are to succeed. The key to the project's success lies in making improvements based on the needs of the company, taking into account the technologies and methodologies used and establishing an efficient communication mechanism between the members of the organisation. The development tasks are automated by requesting a registration of activities with information on pending issues [112].

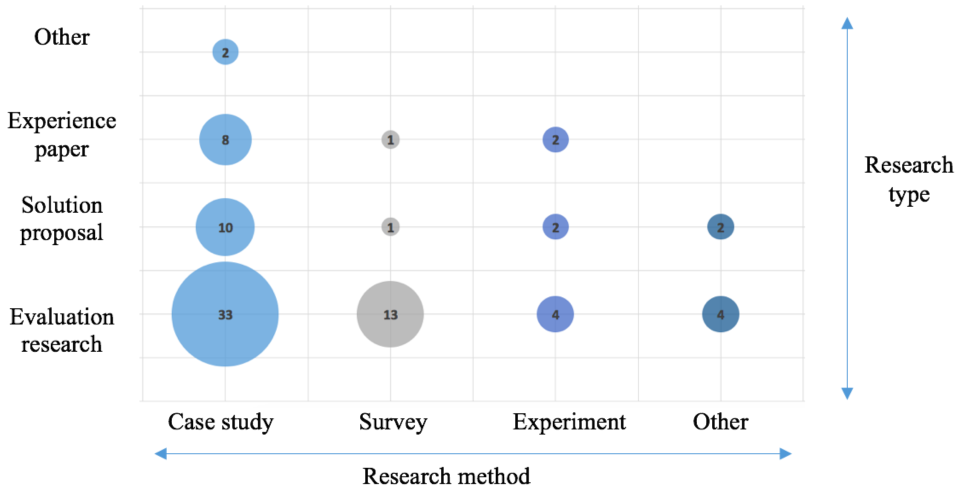

FIGURE 3.5: Research methods and research types

3.5.6 Research type of SPM for GSD papers (MQ5)

Fig. 3.5 shows the type of research undertaken in the selected papers. The most frequent type is evaluation research at 70.5%, followed by solution proposals. Three papers in the SMS present only a qualitative evaluation by the provision of guidelines and theory but no empirical data. Although there is a predominance of solution proposals and experience reports published from 2000 to 2007, the focus has recently shifted to the validation and evaluation of SPM approaches in the GSD setting, which indicates that the main concern of researchers regarding SPM in the GSD domain is to evaluate and validate techniques with which to enhance SPM for GSD.

This result indicates that the SPM for GSD domain is not yet a mature field [88], particularly with respect to testing hypotheses and utilizing approaches and tools. Researchers are still seeking approaches that will enhance SPM for GSD and are generating papers in which software engineering methods, techniques, and tools are evaluated in industrial environments where GSD is practiced [118]. The typical challenges of GSE have been identified as task coordination and allocation, collaboration and knowledge management, and resource estimation and effort [75], [119]. These problems are related to the communication and coordination between global teams, team performance, and development costs. To solve these issues, an approach must utilize steps with which to define and standardize work processes across sites, conduct project retrospectives, and describe work processes [120].

3.5.7 SPM activities for GSD (MQ6)

Systematic software development is a process in which team members with different skills perform different activities [117]. We classified the selected papers according to the activities in which they were mainly involved. To do so, we used the following PMBOK process groups: planning, monitoring and controlling. These process groups include the following knowledge areas: (1) Project integration management (19,6%), (2) Project scope management (15,6%), (3) Project time management (11,8%), (4) Project cost management (11,8%), (5) Project human resource management (19,6%) and (6) Project communication management (21,6%).

Table A.7 of Appendix A provides a summary of these activities, in which the second column specifies the activity of each knowledge area. The fourth column lists the papers in which these activities are identified. The most important finding is that 21.6% of the selected papers address project communication management issues, which indicates that communication plays a critical role in the success or failure of a GSD team [8].

3.5.8 Benefits and limits of the use of SPM approaches in GSD (MQ7)

The literature on SPM mentions many approaches related to GSD. Each approach has potential benefits which can serve as successful practices to facilitate the management of distributed development. Table A.8 in Appendix A provides a summary of the benefits and limits related to these approaches.

Agile based management methods have been successfully used in distributed projects [121]. They emphasise explicit communication and provide many useful communication practices. Many papers deal with methods related to agile approaches: eXtreme Programming (XP), Scrum, Feature-Driven Development, and Crystal. Some common goals of these methods are to help coordination between parties and to lessen the reliance on heavy upfront planning [122].

The 'model' approach includes: (1) the constructive cost model (COCOMO), which minimizes the language gap, (2) the capability maturity model integration (CMMI) model [81, 123], which identifies best practices that help collaborative organizations to improve their processes and (3) estimation models [112] to enable the recording of difficult but necessary decisions made in the management of GSD to guide the project to success. It is not possible to correlate these estimation models with the number of locations involved in a distributed project. The results based on an analysis of the data gathered by the selected papers indicate that estimating the effort expended in GSD projects remains difficult. Although the results do indicate that some models were determined to be more appropriate than others, these findings must be analyzed with caution, as they do not enable researchers to use existing metrics to validate their perceptions about the models' estimation accuracies.

The data mining techniques employed can apply GAs to identify project purposes. This approach also includes CBR [124], which is a technique used to resolve new problems based on the solutions identified for analogous past problems. CBR is therefore a solution option that eases the problems associated with the recovery and application of knowledge in GSD. Unlike GA technology, the CBR methodology indicates how to solve problems at a particular time using previously stored solutions without the need to specify a particular technology.

The Delphi process also provides responses to difficult decisions in GSD management [125], such as choosing coordination mechanisms and tools, selecting a methodology, and assigning work packages. Table A.8 also highlights the benefits and limitations of the main frameworks. Despite the limited predictions of RBV, this framework provides a wealth of in-depth information on offshore application development.

3.6 Discussion

In this section, we summarize and discuss our main SMS findings and identify some implications for researchers and practitioners.

3.6.1 Principal results

The goal for this SMS is to provide an overview of the current literature on SPM approaches for GSD, evaluating the quality of the papers studied and explaining specific findings according to the seven criteria previously indicated.

- **MQ1 and MQ2**. The study confirms that SPM for GSD still have a high level of importance and attracts the attention of researchers and practitioners. The first study was published in 1998 [117]. The studies have recently been specifically published in journals and conferences, especially the ICGSE conference, and the most productive period of SPM for GSD began after the appearance of the first edition of this conference in 2006. This shows that the progress in the field of GSE started with outsourcing at the end of the last century, which led companies to build international corporations and choose to immigrate to the multi-national companies with development sites in different countries [88].
- **MQ3**. Literature usually considers the importance of decision support in distributed project management in the context of task allocation. A careful plan must be elaborated if the management of the combined use of GSD and agile methods is to succeed. The development of complex products using Agile Development and its methodologies began in the 1990s [113]. These agile methods have also been used for GSD since they help solve complex problems by effectively relying on frequent communication and quick feedback and also reduce the emphasis on documentation. In the GSD context, an agile team is a cross functional team that has, in its entirety, the complete collection of skills needed to perform software development activities and deliver values to customers.

  The outcomes of the SMS show that the main inspiration for the research into SPM for GSD has been studies from software engineering research on distributed software development. In fact global software projects can be categorized in two dimensions [126]: Relationship structure and geographic location of work. The former refers to the development of software in an outsourcing versus insourcing arrangement, while the latter concerns the accomplishment of the project, which is either carried out offshore (located in a different country to that of the client organisation), or onshore (located in the same or a nearby country). The types generated by these two dimensions determine a simplified matrix of GSD business models [127]. The selected papers mainly (90%) deal with the offshore case and outsourcing, which are the existing pillars of the matrix in our case.

- **MQ4**. Seventy percent of the selected papers were case studies, but only 9% of them were experiments. This low percentage can be explained by the extra effort required to evaluate goals in global settings. GSD research is hampered by distance and the difficulty involved in finding suitable industrial projects [128]. This result hints that there may be a lack of collaboration between software companies and researchers. Moreover, only 4% of the selected articles included industrial experiments in their research. What is more, the authors of 15% of the selected papers used surveys to collect quantitative information about SPM approaches for GSD .

- **MQ5**. The earliest evaluation research selected in this SMS was from 2006, but the real interest in evaluation research did not start until 2010 [88]. Evaluation research became the dominant tendency in literature owing to the existence of previous approaches for SPM that were used to evaluate, validate and compare [79].

- **MQ6**. To establish a finding, well-known modelsmust be taken into account. The PMBOK and SWEBOK knowledge areas constitute classification models and analysis with which to best match knowledge areas in software engineering management [75, 88, 129]. Few researchers have based their studies on SPM BOKs and standards, yet these BOKs and standards represent a lucid, precise and detailed structure for analyzing results and putting work in its appropriate context. The flow of information among groups of processes in BOKs and standards should be taken into account by researchers who wish to see SPM for GSD succeed [75].

- **MQ7**. GSD has become a dominant paradigm in the software industry [80]. The needs as regards communication, collaboration and knowledge management among team members have led researchers to propose approaches that are used for these purposes. The benefits have been deemed to outweigh the challenges owing to the impact of one particular benefit [130], "diverse knowledge and market proximity". Other benefits include an improved understanding of agile practices and ways of working, better teamwork, higher product quality, and lower overall project cost [131]. It was found that this benefit was also critical to the successful performance of the GSD project.

  In an outsourced project, the outsourcer assigns the responsibility of project management to the project manager from the outsourcing country [123]. That project manager is responsible for project planning, risk management, time management, team management and other issues related to the project. CMMI models have been proved to minimise and prevent risks, especially in processes that deal with outsourcing vendors [75]. Various approaches with which to facilitate offshore work and to achieve good management for GSD are presented: eXtreme Programming (XP), the Resource Based View (RBV) and its extension "Dynamic Capabilities". XP methodologies [132] reduce the communication delay and improve communication quality. RBV [133] helps identify the key project management capabilities associated with offshore application development. The Dynamic Capabilities framework extends RBV by adding a time-based capacity in order to either renew or improve the resources so as to better meet the needs of a changing competitive environment. Our results indicate that in order to fully attain the benefits of offshore outsourcing, there is a need to know the nature of the project, identify important inter-organisational challenges that distributed projects confront during the entire software development life cycle and identify the most effective approaches that will probably be needed. The limitation of direct communication in GSD makes the need of effectiveness on indirect communication. Although indirect communication can vary depending on the used approach, COCOMO, Flow mapping and QAP minimise and make it easy to plan and manage communication. Procura and GA are two approaches that helped the project manager provide optimised planning and scheduling in a hierarchical top-down approach. There are two sections that need to be covered, which are effective guidelines and iterative process of modeling. Both of them are done in Multi-criteria decision model and VTManager method but the question is how to make them become live documents. For that purpose, a template document can be adapted to different circumstances as the artifact in global software projects. The document layout consists on changing control and revision section, introducing the document, releasing planning section, complexity estimation section, iteration planning section, and summary section. The template is developed in casual workflow as like as we construct software in agile methods [134]. Estimation models provide information about the estimation for each feature to be developed. It can line of codes (LOC) estimation, user stories estimation, or simply as development hours estimation

3.6.2 Implications for researchers and practitioners

The outcomes of this SMS have implications for researchers who are working in the SPM for GSD domain, since this paper will allow them to discover the approaches existing in literature. With regard to the implications for practitioners, this review has found that few experimental studies have been conducted in industrial environments. This evidence may imply that the application of SPM for GSD in industry is quite limited. We accordingly suggest that practitioners might cooperate with researchers to investigate the chance of applying the new approaches in their practices. These findings indicate that the field is still quite immature in terms of its being more problem-oriented than focused on solutions and particularly on empirically evaluated solutions. Furthermore, complementary information is provided by means of the following recommendations.

Practitioners such as software developers, project managers and researchers involved in GSD project management should read articles published in the proceedings of the ICGSE conference and its affiliated workshops, in addition to journals that specialise in research and experience, thus contributing to the improvement of software development practices in general: The Software IEEE journal provides issues and practices, and includes methods and techniques with which to better engineer software and manage its development, while the Empirical Software Engineering journal is useful as regards searching for empirical research published in scientific journals specialising in the software project domain. This list above represents the main publication sources for studies related to SPM for GSD. This same list presents sources to which researchers are encouraged to send their articles.

A much smaller group of papers discusses successful practices and shares the lessons learned from them [118, 135, 136]. Practitioners could therefore first take these practices into consideration as a useful path to follow as regards adopting, constructing and developing SPM approaches for the GSD context [137]. Researchers may benefit from this paper by choosing the SPM for GSD approaches that best fit their needs. More studies involving recent SPM for GSD approaches are needed in order to develop skills that will meet distributed software challenges, particularly the need to focus on this kind of development as a part of the future of the software development business.

The result of this SMS shows that SPM for GSD related subjects would appear to need more investigation by researchers, particularly approaches such as solutions to deal with the challenges related to contextual information management, knowledge management and performance management in GSD [80]. Researchers should carry out more research into distributed software project activities, since communication, coordination and the cost of software applications become increasingly more difficult as the project size increases [21]. The best example that can be used to give an idea of the utility of an approach and its classification is that researchers may obtain benefits from the result of the mapping question that deals with the benefits and limits of SPM approaches for GSD.

3.7 Threats to validity

Below, we discuss the potential threats to validity and the steps that have been taken to mitigate or minimize these threats [138].

- *Construct validity:* The construct threats to validity in an SMS are related to the identification of primary studies [139, 140]. A high quality SMS should be based on a stringent search process. A carefully designed search string has been proposed using an extensive range of terms in order to obtain an exhaustive list of relevant primary studies. Different terms exist to introduce the key words. Consequently, the results found might not be complete. This can be attributed to either the search process or the fact that other terms were missed out of the search string which may have affected the final list of papers selected. However, we iteratively built the search string and performed a systematic search using an extensive range of terms to widen our research scope. The identification of these search keywords can be identified as a threat to construct validity.

  The search was performed by using IEEE Digital Library, ACM Digital Library, Science Direct, SpringerLink and Google Scholar. Although this may represent a threat to validity, the primary studies identified (84 papers), the information retrieved and the papers analyzed published in the main conference on GSE (ICGSE) appear to be sufficient to gain a profound understanding of the topic investigated. The references in the selected studies were not scanned to identify further studies. Nonetheless, an important number of articles (37968) were identified through databases. All these papers were checked by at least reading the title in order to identify relevant papers.

- *Internal validity:* Internal validity deals with extraction and data analysis [139, 140]. Two authors performed the data extraction and classification of the primary studies, while the other authors reviewed the final results. The decision as to which data to collect and how to classify the papers was therefore performed on the basis of the judgement of the authors conducting the SMS. These authors, who were from different cultures and research groups, carried out two different classifications for reasons of reliability [102]. This threat has been mitigated by using the Kappa coefficient, thus achieving a score of 0.9, which represents a high level of conformity. This threat is therefore minimal, and only has a minor influence on the general classification derived from this study.
  When conducting the SMS, articles were not excluded on the basis of a quality assessment. Although some researchers might find it preferable to exclude those

articles, including them clarifies and develops the results of our SMS and allows us to enrich our discussion.

- *Conclusion validity:* In the case of an SMS, this threat refers to factors such as missing studies and incorrect data extraction [139, 140]. The aim is to control these factors so that an SMS can be performed by other researchers who will draw the same conclusions. Bias both as regards selecting and classifying primary studies and analyzing data may therefore affect the interpretation of the results. In order to mitigate this threat, every step performed in the selection and data extraction activity was clearly described as discussed previously. The traceability between the data extracted and the conclusions was strengthened through the direct generation of charts and tables from the data by using a statistical package. In our opinion, slight differences based on publication selection bias and misclassification would not alter the main conclusions drawn from the articles identified in our SMS. The threat to conclusion validity is thus covered.

- *External validity:* External validity is related to the generalization of this study [141, 142]. This SMS considers the GSD domain, and the validity of the conclusions drawn in this paper concerns the GSD context. This threat is not therefore present in this context. The results of this study may serve as a starting point for SPM for GSD researchers, and practitioners can search for and categorize additional papers.

3.8 Conclusions

This chapter is an SMS that analyses the literature related to SPM for GSD and the aspects contributing towards the success of GSD projects. The status of SPM approaches applied in the GSD arena is now briefly described by means of a SWOT analysis [143]. Strengths, weaknesses, opportunities and threats of the SPM approaches for the GSD arena, are the four factors that justify the use of SPM approaches within the scope of GSD, the issues to be resolved, the profitable areas of development, and the measures to be addressed by the SPM for GSD community.

The baseline is one of the most relevant strengths. It is a quite robust and mature strength with a large amount of empirical evaluation of methods, techniques and tools in an industrial context. It provides and combines multiple data sources for cross-validation. Furthermore, it explores the issues raised by the initial studies. In addition, SPM for GSD is a productive field where achievements are disseminated and new research projects are encouraged.

With regard to weaknesses, SPM for GSD is an emerging research field with problems that are borne on few shoulders. Most of studies were not published until 2006, with the appearance of the ICGSE conference that specializes in distributed projects. Most of the SPM for GSD approaches represent the implementation of processes exploring theoretical issues, rather than contributing towards extending the field.

Opportunities for SPM for GSD are huge. Software developers are using software engineering management approaches to deal with proximity and inadequate process. Developers have proposed approach usage patterns that are not enforced by processes. They are adapting the approaches and applying social interaction processes and emergent processes, thus making the approaches available to meet their coordination needs.

Threats consist of the constraints that prevent, delay, and obstruct the rational and formal development of SPM for GSD. These obstacles are mainly represented by the natural shortages that an incipient discipline has to confront. SPM for GSD must therefore confront the lack of a relationship between academy and industry. An extra effort should be developed to make the transfer of technology from academy to industry possible. An additional threat to the topic is the lack of visualization and dissemination of contextual information based on the objective, profile, context and format of the software project.

The results obtained from this SMS have led us to a global view of a relatively new topic which should be investigated in detail. However, each distributed project has real needs which basically depend on its distribution characteristics, its activity and the approaches it employs. These factors make this subject extremely large and complex, leading to the need to adapt both technical and organizational procedures according to each of the specific needs of GSD.

Chapter 4

Overview of software cost estimation for GSD

4.1 Introduction

Global Software Development (GSD) refers to software work starting at geographically separated areas across national boundaries considering synchronous and asynchronous interaction. GSD has been adopted by numerous companies. However, these global projects confront a number of problems, which are particularly linked to the gap between different participants: physical distance between the groups of developers causing a lack of trust, time-zone differences, communication problems among teams, effort estimation problems, cultural differences, and others. Current research tends to characterise these problems, but if success is to be achieved in GSD, companies must minimise challenges by adjusting their processes and rearranging their tools and organisational structure.

GSD projects can increase requirements as regards development processes, project management practices, architecture, quality, collaboration tools and so on. These challenges may exceed the advantages of the lower labour rates in the developing country since they could lead to substantial overheads in the day-to-day operations of a GSD project. This reasoning shows that it is vital to understand and estimate the total costs of GSD in order to help evaluate the comparison with local software development in terms of efficiency.

A large range of software cost estimation techniques had already been developed before the GSD trend began [144]. Early research on the topic was conducted in 2006 [25], when researchers were able to promote analyses of project factors in order to gain insights into the comparison of development costs for distributed software development projects and collocated projects. In a study published in 2012 [145], researchers advance the question of cost estimation for distributed software projects by identifying challenges and proposing solutions with which to better drive estimates. Britto et al. [91] present a systematic literature review on effort estimation in GSD. In their study, only 5 papers were selected, which allowed the extraction of only 10 estimation methods. It is important to note that the study of Britto et al. did not consider software maintenance effort/cost estimation; it only concerned software development effort/cost estimation. What is more, the study did not classify the techniques according to their contribution type. Considering the importance of the above limitations, the objective of this chapter is to carry out a systematic review which: 1) includes 16 selected papers among them the 5 ones of Britto et al., 2) considers software development as well as maintenance effort estimation, and 3) discusses effort/cost estimation performance.

The chapter is structured as follows: Sect.4.2 presents the research method used in the study. Sect.4.3 reports the results and findings obtained from the SMS. Sect.4.4 outlines threats to validity. Sect.4.5 discusses the main findings and presents implications for researchers and practitioners, while our conclusions are presented in Sect.4.6.

4.2 Research methodology

4.2.1 Mapping questions

The SMS was performed to obtain the current research on software cost estimation for GSD. This study answers nine mapping questions (MQs). These questions are presented in Table 4.1.

4.2.2 Search strategy and paper selection criteria

The articles were identified by consulting the following sources: IEEE Xplore digital library, ACM digital library, ScienceDirect and Google Scholar. The following search string was used in order to perform the automatic search in the digital libraries selected:

(Software **OR** system* **OR** application*) **AND** (cost **OR** effort **OR** resource) **AND** (estimat* **OR** plan* **OR** predict* **OR** measur* **OR** calcul* **OR** manage* **OR** control*) **AND** (Global development **OR** distributed development **OR** outsourc* **OR** Offshor* **OR** Dispersed development). This search string was applied to the title, abstract and keywords of the papers to reduce the search results. Each paper was retrieved by the first author and specific information of each relevant paper was filled in an Ms Excel file.

Table 4.1: Mapping questions

ID	Mapping question
MQ1	Which publication sources and channels are the main targets for software cost estimation for GSD research?
MQ2	How has the frequency of software cost estimation for GSD research changed over time?
MQ3	What are the research types of software cost estimation for GSD studies?
MQ4	Which research approaches are used in software cost estimation for GSD studies?
MQ5	What are the contribution types of software cost estimation for GSD research?
MQ6	Which cost estimation techniques are most frequently used for GSD projects?
MQ7	Which software cost estimation activities have been addressed by GSD research?
MQ8	Which cost drivers affect GSD projects?
MQ9	Which cost estimation performances have been obtained from GSD projects?

The aim of the selection process was to identify the most relevant studies for this mapping study. Each paper was retrieved and evaluated by one author who decided whether it should be included by considering its title, abstract and keywords. The final selection result was reviewed and approved by the remaining authors. The first step after the articles had been identified was to eliminate duplicate titles, and titles which were clearly not related to the review (16 selected studies out of 103 relevant studies). The inclusion criteria were limited to those studies that focused on software cost estimation for GSD projects, and any studies that met at least one of the following exclusion criteria (EC) were excluded:

- EC1: Papers that are not published in journals, conferences or workshops.
- EC2: Papers that are not in English.

4.2.3 Quality assessment (QA) process

The QA in an SMS is a major focus that increases the depth of a study. In order to enhance our study, a questionnaire was therefore designed to assess the quality of candidate papers. The scoring used in this questionnaire was determined on the basis of previous studies [104, 146, 147].

(a) The paper has been published in a recognized and stable journal or conference. This question was rated by considering the computer science conference rankings in the Computing Research and Education (CORE) 2013 Conference Rankings, and the 2013 Journal Citation Reports. The possible answers to this question were:

- For conferences: (+2) if it is ranked CORE A*; (+1.5) if it is ranked CORE A; (+1) if it is ranked CORE B; (+0.5) if it is ranked CORE C; (+0) if it is not in CORE ranking.
- For journals: (+2) if it is ranked Q1; (+1,5) if it is ranked Q2; (+1) if it is ranked Q3; (+0.5) if it is ranked Q4; (+0) if it is not in JCR ranking.

(b) The main focus of the paper is software cost estimation activities used to deal with GSD challenges. Yes (+1); Partially (+0.5); No (+0)

(c) The study is complete and discusses the results obtained. Yes (+1); Partially (+0.5); No (+0)

(d) The study is empirical and presents relevant results for our SMS. Yes (+1); No (+0)

4.2.4 Data extraction strategy

The data extraction strategy was based on providing the set of possible answers to the MQs. The strategy is explained below:

MQ1: In order to answer this question, it is necessary to identify the publication source and channel for each paper.

MQ2: In order to discover the publication trend, the articles should be classified per publication year.

MQ3: A research type can be classified into the following categories:

- Evaluation research: existing software cost estimation for GSD approaches are implemented in practice and an evaluation of them is conducted.
- Solution proposal: a solution for software cost estimation for GSD is proposed. This solution may be a new software cost estimation for a GSD approach or a significant extension of an existing approach.
- Other, e.g. opinion paper, experience paper.

MQ4: The research approach can be classified as being:

- A case study: an empirical inquiry that investigates a software cost estimation approach for GSD within its real-life context.
- A survey: a method used to collect quantitative software cost estimation for GSD information.
- A experiment: an empirical method applied under controlled conditions in order to observe its effects on software cost estimation in the GSD context.
- A review: an analysis of software cost estimation for existing GSD literature.

MQ5: A contribution can be classified as being:

- A technique: a procedure used to accomplish a software cost estimation for a GSD task. e.g. a data mining technique.
- A model: a representation of a system that allows software cost estimation for GSD properties to be investigated.
- Other, e.g. process, tool.

MQ6: Several cost estimation techniques for GSD projects have been used in the last few decades. These techniques can be classified as [58, 148]:

- Expert judgment: this involves consulting a group of experts in order to use their experience to propose an estimation of a given project [149].
- machine learning models: Approaches that are based on soft computing such as artificial neural networks, fuzzy logic models and genetic algorithms [150].
- non-machine learning models: these provide linear and non-linear regression models to establish equations with which to perform software estimation. [151].

MQ7: Software cost estimation activities that were addressed by GSD research can be categorised as:

- Software development cost: Performed by managers and software system engineers for activities such as functional design, software requirement, development code, development tools, integration of software and finally the test procedures.
- Software maintenance cost: Related to the control and the monitoring of the software after it has been delivered to the final user, since there will always be problems with the software as it gets older

- Other, e.g. Reengineering cost

MQ8: Cost drivers that affect GSD projects are divided into 4 categories, namely Product, Platform, Personnel and Project factors [49].

MQ9: In this paper, we focus on the two main criteria that affect GSD projects: geographical and temporal challenges and their influence on cost performance. According to the PMBOK Cost Management knowledge area, cost performance is included in three main outputs of a GSD project:

- Cost Performance Baseline: An authorised time-phased budget at completion used to measure, monitor, and control overall cost performance on the project.
- Work performance measurements: The calculated cost variance for work packages and control accounts
- Basis of estimates: The amount and type of additional details supporting the cost estimate vary according to the application area.

4.3 Results

4.3.1 Quality Assessment

SMSs generally emphasise the quality of selected studies. This QA is usually carried out to discover the general view of the paper's implication in the subject. However, Kitchenham et al. [16] specify that even if some researchers use QA as a selection criterion in their systematic review, this assessment is not mandatory for an SMS. Table 4.2 provides information about the total score of the selected studies. The majority of the selected papers (66.25%) have at least a medium score for quality, which shows that they contain useful information, particularly as regards information on software cost estimation and the impact of GSD projects on the cost estimates. No studies were discarded from these inputs during the QA process.

Table 4.3 shows the number of articles based on the ranking of the conference or journal at/in which they were published.

4.3.2 MQ.1: Source and Channel of Publications

Table 4.4 provides a schematic representation of publication channels and the number of articles per publication source. Table 4.5 presents the journals and conferences at

TABLE 4.2: Quality levels of relevant studies

Quality level	Papers	Percent (%)
Very high ($4 < score \leqslant 5$)	1	6,25
High ($3 < score \leqslant 4$)	1	6,25
Medium ($2 < score \leqslant 3$)	7	43,75
Low ($1 < score \leqslant 2$)	5	31,25
Very low ($0 < score \leqslant 1$)	2	12,50

TABLE 4.3: Articles by their journal or conference rank

Journals	Number	Conferences	Number
Q1	1	CORE A*	1
Q2	2	CORE A	1
Q3	0	CORE B	0
Q4	1	CORE C	6

which the papers selected for this SMS were published. This result clearly shows that the International Conference on Global Software Engineering (ICGSE) is the main publication source for our topic. With regard to journals, systems, software and computer science journals are the targets of researchers in the field.

TABLE 4.4: Publication channel

Publication channels	Selected papers	Percent
Conference	11	68,75%
Journal	5	31,25%
Total	16	100%

4.3.3 MQ.2: Publication Distribution Per Year

Fig. 4.1 shows the number of publications per year. The amount of publications interested in software cost estimation for GSD projects has increased since 2006. This year corresponds to the outset of the increasing concern about the effect of globalization on the software industry in general [89] and is also the year in which the first ICGSE conference took place.

Table 4.5: Journal (J) and Conferences (C) of selected studies

Publication channels	Type	Total
International Conference on Global Software Engineering (ICGSE)	C	4
Computer Science and Information Technology (CSIT)	C	2
Software Engineering International Conference (ICSE)	C	1
SRII Global Conference (SRII)	C	1
Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (ACIS)	C	1
Services Computing, IEEE International Conference (SCC)	C	1
Innovations in Information Technology Conference	C	1
International workshop on Economics driven software engineering research (EDSER)	C	1
IEEE Software	J	1
Advances in Software Engineering	J	1
Systems and Software	J	1
European Journal of Scientific Research	J	1

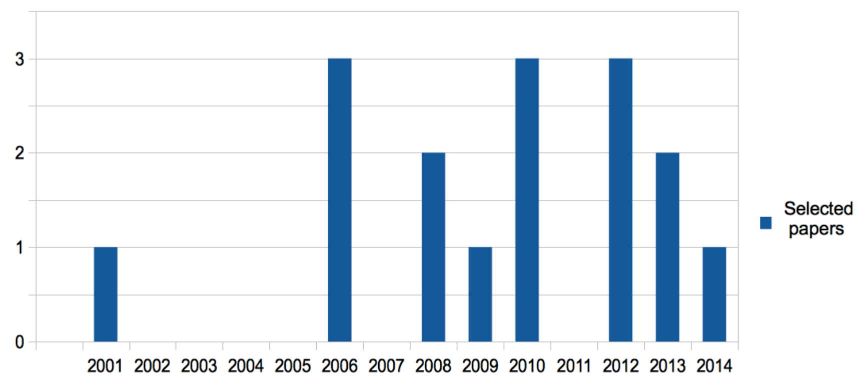

Figure 4.1: Publication per year

4.3.4 MQ.3: Research Type

Sixty two percent of the selected articles are evaluation research, while 25% of the selected papers are solution proposals. 12,50% are contained in the "Other" category, which comprises theoretical papers and experience papers. Fig. 4.2 divides the selected articles by their publication date. The first column shows articles published before the year 2009, the second shows those published between 2009 and 2011 and the last shows those published since 2012.

According to the data shown in Fig. 4.2, the number of evaluation research papers is low in comparison to the number of solution proposals up until 2009. In this period,

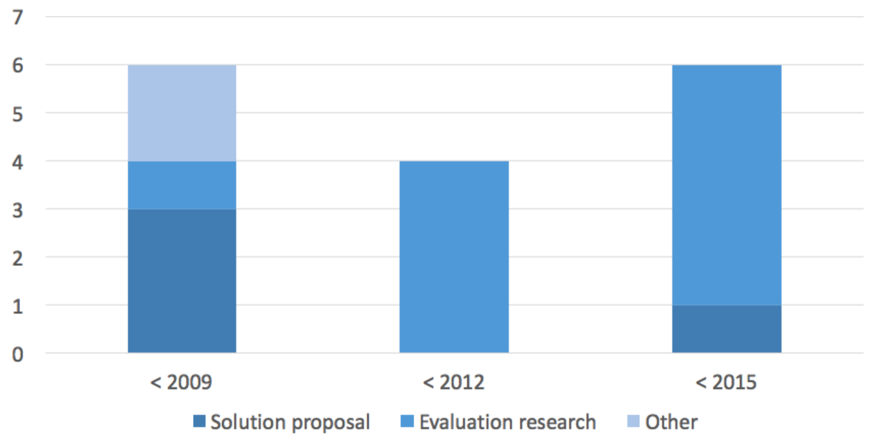

FIGURE 4.2: Research Type

estimation cost for GSD projects was a relatively new subject that needed more investigation and exploration. From 2012 on, the focus shifted to the validation and evaluation of existing software cost estimation methods for GSD projects.

4.3.5 MQ.4: Research Approach

Four of the selected papers are solution proposals. Two of them were validated using experiments, while the other two were not empirically validated. Five out of 10 of the selected evaluation research studies were based on theoretical approaches and 4 out of 10 were based on case studies, while not a single article was based on an experiment. This situation may result from the difficulty involved in observing the effects of methods on software cost estimation under controlled conditions, particularly in the case of distributed projects in the GSD context. More details on MQ.3 and MQ.4 are provided in Table 4.6.

TABLE 4.6: Research Types and Approaches

	Case study	Experiment	Survey	Theory
Evaluation	4	0	1	5
Solution	1	1	0	2
Other	1	0	0	1

4.3.6 MQ.5: Contribution types

Fig. 4.3 presents the distribution of the selected studies' contribution types. Thirty eight percent of the categories are models. Techniques [120] for the cost estimation of GSD represent 50% of the selected studies' types. These models gather data mining techniques used to help researchers and software companies establish data by providing a number of algorithms and methods with which to deal with software cost and effort challenges. Each of these data mining techniques analyses data and provides results with which to best match the software cost of a GSD project. The third partition "Other" represents only twelve percent split between processes and tools.

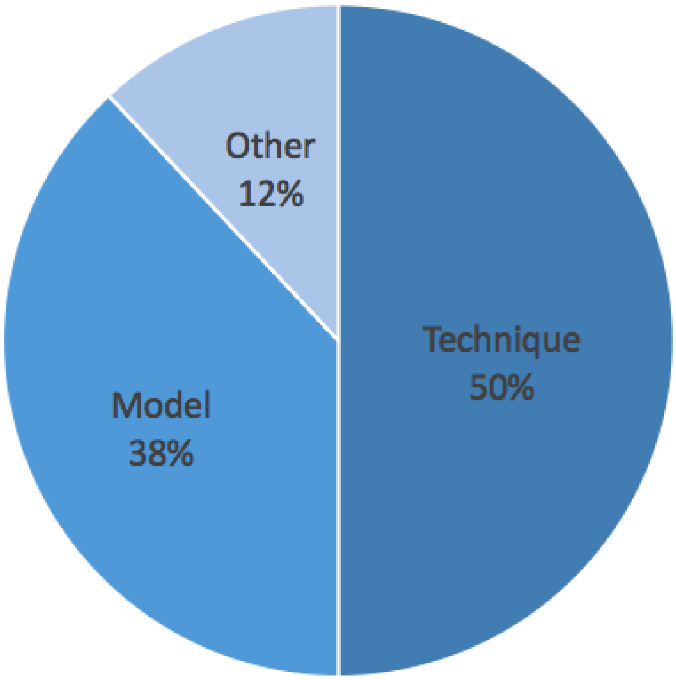

FIGURE 4.3: Contribution types

4.3.7 MQ.6: Software cost estimation techniques

Several software estimation techniques can successfully be used to estimate costs. non-machine learning models (7 out of 16 selected studies) include COCOMO (II) [25], SLIM [28], Use Case Points [29] and Function Points [112]. The use of expert judgment (3 out of 16 selected studies) consists of asking the opinion of multiple experts who use their experience and knowledge of the project to provide an estimation of the cost. An objective estimation is secured by obtaining as many values as possible from different experts. Indeed, the objective of the Delphi technique [112] is to repeat the estimation process until an agreement is established. Table 4.7 presents details of the cost estimation techniques used for GSD projects.

4.3.8 MQ.7: Software cost estimation activities

About 80% of the selected studies discuss the software development costs of GSD projects. These studies show the strong link between the software life cost and its development phase. The second most frequent topic after development cost is contained

TABLE 4.7: Estimation techniques for GSD projects

Type technique	Estimation techniques	Papers	Percentage
non-machine learning models	COCOMO(2)	[25, 27–29, 145, 152]	85,7%
	SLIM	[28, 152]	28,5%
	Function points	[112, 152]	28,5%
	Use case points	[29, 112, 152]	42,8%
	Multiple Linear Regression	[152]	14,2%
Expert judgment	Delphi	[112, 152]	28,5%
	ISBSG	[28, 152]	28,5%
	Planning Pocker	[112]	14,2%
	Epert Judgement	[112]	14,2%
machine learning models	Artificial Intelligence	[153]	14,2%
	Case-based reasoning	[145, 154]	28,5%
	Regression trees	[153]	14,2%
	Neural Network	[152]	14,2%
	Genetic Algorithm	[153]	14,2%

in the few studies that concentrate on the maintenance phase. This phase focuses principally on the extraction and consideration of factors that affect software maintenance. If the software maintenance cost is to be properly applied, it is essential to estimate the cost and reduce it by controlling certain factors.

Table A.8 summarises the elements that affect software cost development and maintenance in the GSD context. Software development and maintenance are the major issues to affect a GSD project. The study is based on the analysis of data collected from selected papers. Software activities have been shown to have significant costs. The development cost resulting from the overall estimate and the estimation of the benefits of strategies and the networking remain highly uncertain and open to improvement, as do the costs incurred as the results of maintenance, particularly modification, improvement or enhancement along with reengineering costs. These costs have been known to erode whatever benefits the GSD model may provide.

4.3.9 MQ.8: Software cost drivers

There are 16 cost drivers, which are divided into the four categories depicted in Table 4.9: Product, Platform, Personnel, and Project Factors. As this table, shows some

Table 4.8: Software cost estimation activities

Activities	Elements	Descriptions	Papers
Software development cost	Development budget	Costs resulting from the overall estimate to software development	[29, 152]
	Software improvement	Costs resulting from estimating the benefits of strategies such as tools, reuse, and process maturity	[145]
	Project planning and control	cost of schedule and control breakdowns by component and activity	[27, 155]
	Project constraints	Costs resulting from the networking, communications, delay in Response and different Time Zone	[28, 112]
Software maintenance cost	Corrective maintenance	Costs resulting from the modification of software into correct issues detected after initial deployment	[145]
	Adaptive maintenance	Costs resulting from the modification of a software solution to help it stay effective in a changing business environment	[145]
	Perfective maintenance	Costs resulting from the improvement or the enhancement of a software solution to improve overall performance	[145]
Reengineering cost	Enhancements	Costs resulting from the sequence innovations	[26]

cost drivers are common to all types of software projects while others are specific to GSD projects. The majority of cost drivers that impact on GSD projects are related to factors in distributed software projects: time zone, language (communication) and cultural differences (team culture). Note that the most frequently used cost drivers are project effort (44,4%), process model (33,3%) and time zone (22,2%).

In order to establish trust in distributed projects, researchers recommend bringing about cultural understanding, creditability, capabilities, pilot project performance, personal visits and investments in the field of GSD. These studies also suggest cultural understanding, capabilities, contract conformance, quality, timely delivery, development processes, managing expectations, personal relationships and performance as the key factors for better achievement particularly as regards good communication.

Table 4.9: Software cost drivers

Category	Cost drivers	Papers
Product	Code size	[28]
	Reuse	[145]
	Product complexity	[25]
Platform	Design and technology newness	[26, 145]
	Time zone	[27, 155]
	Platform volatility	[25]
Personnel	Team size	[145]
	Team culture	[29]
	Team trust	[28, 29]
	Communication	[25, 155]
	Development productivity	[27, 145]
Project	Project effort	[27, 28, 112, 145]
	Project management effort	[27, 112, 145, 152]
	Process model	[28, 29]
	Task allocation	[27, 155]
	Work Pressure	[28]
	Client involvement	[145]
	Work dispersion	[25, 145]

4.3.10 MQ.9: Software cost performances

The main reason for studying cost performances in the GSD context is to reduce costs. Five different cost performance variables were included to quantitatively characterise GSD projects: Distributed work, client control and behavior, project team, project methodology and technology variables [145].

Cost performance is principally evaluated in three ways, as can be seen in Table 4.10 [145]. The direction of distributed development as regards cost performance is decided by the direction of the methods in a statistical test model created using quantitative data or grounded conclusions from qualitative data obtained from primary studies. In summary, only 3 studies provide estimates derived from empirical data obtained from cost performance methods applied in different projects of different companies.

Table 4.10: Software cost performances

Evaluation type	Cost performance method	papers
Baseline comparison	Historical project databases	[28]
Variation reduction	MRE , Prediction level	[145]
sensitivity analysis	CCNN	[152]

4.4 Threats to validity

The results of this SMS may have been influenced by the coverage of the study search, bias in study selection, and inaccuracy in study data extraction. Four types of threats to the validity [156] of the study results are therefore discussed in the following subsections.

Construct validity is concerned with the exactitude of the interpretation of the concepts studied and the completeness of the relevant studies collected. In this mapping study, the key concepts under consideration are contributions towards software cost estimation for GSD projects. To ensure the correct interpretation of these key concepts, we verified the definitions of the concepts in related literature and all the authors discussed these definitions in order to reach a consensus as to their understanding of them.

Internal validity is concerned with the analysis of the data extracted. The threats to internal validity are minimal considering that only descriptive statistics were used during the data analysis in this SMS.

Conclusion validity is concerned with the search terms used in the automatic search and the search sources are presented in order to make the results of this mapping study reproducible.

External validity is concerned with the representativeness of the selected studies as regard the overall goal of the mapping study. The results of this mapping study were considered with regard to the software cost estimation for distributed projects. These results and representative venues can serve as a starting point for researchers and practitioners working in this field.

4.5 Discussion

This mapping study indicates that the application of software cost estimation techniques for GSD projects is a fairly immature area in both research and practice. First, about two thirds of the selected studies (11 studies out of 16) were published at conferences and workshops, while only 31.2% (5 out of 16) of the selected studies attained the maturity needed to be published in a journal. Furthermore, only one of the selected studies [145] attained a very high quality level (i.e., evidence obtained from QA).

The fact that the number of selected studies increased over the last decade shows that the application of software cost estimation knowledge is receiving increasing attention from the software research community. The selected studies were published at 12 different venues, indicating that extensive attention is being paid to this study topic by researchers with a broad range of different research interests in software cost estimation. All of the above indicates that this study topic is likely to remain attractive. However we would urge the research community to strive for high-level evidence in future studies. The results of this SMS also highlight a number of implications for further research in the field:
(1) Challenges associated with software cost and effort estimation in GSD are not new. One of the main reasons for the growth in GSD is the cost of reducing software development, and effort estimation is a key component of this cost. Good effort estimation is thus important for the success of any GSD project. The results of this mapping study show the need for more research into techniques that can be used to improve software cost estimation analysis. An adaptation of techniques and models that takes into account the challenges and factors associated with GSD must also be investigated.
(2) This mapping study also shows that the application of the knowledge recovery approach in various forms needs to be explored seriously. In many software cost estimation cases, practitioners need to recover the knowledge about software characteristics, especially when developing or maintaining a global software project that is not well described and documented. But little work has been done on the application of knowledge recovery in software cost estimation activities for GSD.
(3) The quantification of the cost drivers' impact on productivity implies a high degree of objectivity and accuracy. However, concepts such as the impact of communication or team trust and team culture on productivity are very difficult to quantify, and the results should be treated with care. This is owing to the complexity and unpredictability of personnel behaviour which has the greatest impact on estimation costs, particularly in distributed development.

4.6 Conclusions

This chapter presents the outcomes of an SMS of cost estimation in the context of GSD projects, in order to serve both research and practice. This SMS has shown a wide spectrum of software estimation techniques, activities and cost drivers for GSD projects. Most of the selected studies present cost contribution as regards cultural, language and time zone differences, which are directly related to making the achievement of globally performed software projects more stimulating.

Upon considering the lack of primary studies identified in this SMS, we believe that further research is required into the approaches used in the GSD context. We are also of the opinion that the adaptation of those techniques based on the specific aspects of GSD, in addition to the inherent uncertainty of the data, could provide more faithful estimates of effort. The globally distributed environment implies many challenges and elements. The GSD sourcing strategy and cost estimation process topology could have a great influence on cost estimates. Future research should therefore be carried out to explore how these challenges and factors affect cost estimation techniques.

Chapter 5

Identifying software cost attributes of software project management in global software development: An Integrative Framework

5.1 Introduction

Identification of software cost attributes is one of the most important processes of SPM and has been widely discussed in literature [157, 158]. Interest in GSD is rapidly growing, identifying the new challenges associated with this phenomenon has push researchers to investigate more to select relevant list attributes adapting to a specific framework and then compiled using survey researchers and practitioners of software project managers involved in GSD.

This chapter gives the investigation of software cost factors used to address the software cost estimation related to the management of GSD projects that are available in literature. The aim of this chapter is to develop a framework encompassing these attributes using a list of software cost attributes that answer the question of "what" is to be built in the GSD context.

In this chapter, we present the results of our study in identifying challenges related to SPM in GSD and categorizing them according to the selected model. The list of software cost attributes has been acquired through an SLR of studies focusing on the approaches facing project managers of GSD projects. To do this, we intend to address the following research question (RQ):

RQ1: What are the software cost attributes of software project management in GSD context?

The chapter is structured as follows. Sect. 5.2 discusses related work. The research methodology is explained in Sect. 5.3. Results from the SLR are presented and discussed in Sect. 5.4. Threats to Validity are presented in Sect. 5.5. Finally, conclusions are presented in Sect. 5.6.

5.2 Related work

One of the earliest researchers in software cost estimation has been carried out by Heemstra [159] prior to the emergence of GSD context. He emphasized the reasons for overruns of cost, and the prerequisites for estimating in the development software life-cycle to lessen software budgets, and avoid software cost estimation challenges. He presented an overview of software cost estimation models with a framework listing the factors which are commonly regarded as important structured in five categories. The emergence of GSD has profoundly changed the way software cost approaches are developed. Some of the factors that are present in collocated software development have a more critical impact on the outcome of GSD projects while they also face new challenges innate to their distributed setting. This change highlights the need of new tailored approaches to the software cost estimation for GSD projects [29].

Keil et al. [25] presented in 2006 an additional cost drivers of distributed development, and examined the significance of each of these factors as a contributor to the overall cost of a software development project. The paper aims to identify 18 effort-multiplier cost drivers related to SPM for GSD. It classifies drivers into four categories (product, platform, personnel, and project factors) and was inspired from previous study carried out by boehm et al. [160].

In 2006, Smite et al. [161] published an article presenting a framework for overcoming threats in global projects. The research aims to investigate the nature of global risks and build a comprehensive framework presenting global factors and threats that distinguish distributed projects. The data was gathered from literature [162], and analyzed according to principles prescribed by a grounded theory through applying open, axial, and selective coding techniques [163]. The study derived several global factors from the performed taxonomy of software development risks, developed by Software Engineering Institute [164]. The results of this chapter show that these frameworks fail to cover all software cost factors presented in literature mainly because the lack of investigation tackled by researchers. In this chapter, we aim first to lay out an updated check-list of software cost factors related to managing GSD projects and then to provide an integrative framework for identifying and classifying these factors.

5.3 Methodology

An SLR is a means of identifying, evaluating and interpreting all available research relevant to particular RQ [16]. One of the most common reasons for undertaking an SLR aim to provide a framework/background in order to appropriately position new research activities. Protocol of this systematic review has been written based on the recommendations of Kitchenham and Charters [16] to describe the review process, three main points are then extracted:

1- Search strategy: Identification of RQs, construction of search string and definition of relevant libraries.

2- Study selection: Conduct the review and perform the selection criteria.

3- Data extraction and analysis: extraction, monitoring and synthesis.

5.3.1 Search strategy

This SLR aims to answer the following RQ:

RQ: What are the relevant software cost attributes in GSD context?

Using the PICO method (Population, Intervention, Comparison, Outcome) on the RQ, the search string have been identified:

- Population: Global Software Development
- Intervention: Cost estimation
- Outcome: Attributes

Synonyms for each of search terms are identified based on previous studies. They are presented in Table 5.1.

The search string used to perform the automatic searches in the digital libraries selected was formulated using the OR logical operator between synonyms and using the AND

TABLE 5.1: Synonyms of the initial search terms

Population	Intervention	Outcome
global software development	cost estimat*	attribute
distributed project	cost plan*	driver
dispersed development	cost predict*	factor
offshor*	cost measur*	
outsourc*	cost calcul*	

logical operator between combined terms of the PICO method. The search string used is:

(Global software development **OR** distributed project **OR** Dispersed development **OR** outsourc* **OR** Offshor*) **AND** (cost estimat* **OR** cost plan* **OR** cost predict* **OR** cost measur* **OR** cost calcul*) **AND** (attribute **OR** driver **OR** factor)

The extracted papers were identified using the specified search terms, and the searches have been performed in the following sources in January 2018:

- IEEE Xplore digital library [1]
- Association for Computing Machinery (ACM) digital library [2]
- ScienceDirect [3]
- SpringerLink [4]

5.3.2 Study selection

This search string was applied to the titles, abstracts, and keywords of papers to reduce the search results. Each paper was retrieved by two researchers separately who each identified relevant studies. When there was disagreement between the two researchers, in a meeting between all the researcher involved in this study, the full text was accessed to reach an agreement. We defined the following inclusion and exclusion criteria based on established SLR guidelines [56]:

Inclusion criteria

[1]http://ieeexplore.ieee.org
[2]http://dl.acm.org
[3]http://sciencedirect.com
[4]http://link.springer.com

- IC1: Papers related to software cost aspect of managing GSD projects.

Exclusion criteria

- EC1: Papers that are not in English.
- EC2: Papers that are published before 2010.

The quality assessment process was not carried out in this SLR. The main reason behind this is to propose an exhaustive list of software cost attributes related to SPM in GSD context found in academic literature.

5.3.3 Data extraction

The data extraction strategy was carried out by the main author in June 2017 and reviewed by the remaining authors. The following data was extracted from each paper by obtaining a set of possible answer to the RQ.

Software Engineering Institute (SEI) taxonomy is developed to support software cost management in different kinds of software development projects, while our aim was to facilitate cost estimation in GSD [165–167]. Taxonomy organizes software cost estimation threats into three major classes: Product Engineering, Project Personnel, and Development Environment. We used a SEI taxonomy to classify the identified and uncovered global software cost attributes. Boehm et al. [168] defines these three categories as follows:

- Product Engineering: focuses on the nature of product engineering and process, product requirements, and project constraints.
- Project Personnel: focuses on the resources and skills for managing the software project.
- Development Environment: deals with how project development tasks can be achieved. Typical concerns are: operating system, software development tools, project time, and how tasks are allocated in developing the project.

5.4 Results and discussion

5.4.1 Study selection results

In total, after the application of the search string to the digital libraries, we identified 268 papers. The list of identified studies in the initial selection phase was comprised of 45 articles. After the full text reading during the final selection phase 12 studies were finally selected. Table 5.2 summarizes the results per each of the digital libraries used.

TABLE 5.2: Search Results

Digital Library	Returned studies	Initial Selection	Final Selection
IEEE Xplore	54	19	6
ACM	32	10	2
Science Direct	179	15	4
SpringerLink	3	1	0
Total	268	45	12

To answer our RQ, the data was extracted and synthesized from the 12 finally selected studies. The final list of cost attributes identified in this study comprises 39 cost attributes related to GSD project. The attributes have been classified into the three categories and are presented in the following subsections:

5.4.2 Cost attributes within "Product Engineering"

The software cost attributes derived from Product Engineering mainly relate to the complexity, reusability and reliability required to carry out the specified product, they are presented in Table 5.3

Nine software cost attributes have been identified to fit into this category, they deal with processes for handling product, description and documentation. The most commonly mentioned software cost attributes in this category is "Reusability". Reusing experiences and lessons learned in distributed software projects help to understand the characteristics and problems of distributed development, which might increase the project planning task.

Table 5.3: Software cost attributes related to "Product Engineering"

Cost attributes	Reference
Reusability	[25, 27, 29, 30, 91, 145, 169]
Complexity of the software	[25, 27, 29, 169]
Portability	[27, 29, 169]
Documentation	[25, 27, 30]
Code size	[25, 145, 169]
Reliability	[25, 27, 170]
Requirement legibility	[29, 91]
Defect density	[91, 145]
maintainability	[27]

5.4.3 Cost attributes within "Project Personnel"

Personnel factors focus on the cultural fit, the closeness of team members and also the experience of the teams. It concerns educational level, language skills and knowledge management aspects of GSD projects. Nine software cost attributes that fall within this category have been identified in literature. They are presented in Table 5.4.

Software cost estimation in GSD projects includes specific cost overheads drivers and site-specific characteristics. The most reported cost drivers that fall into this category are associated to language skills and cultural differences which strongly affect distributed software development. Keil et al. [25] identified formal personnel factors, called "soft skills", to be a key success factor in distributed projects and provided mechanisms to coordinate cross-site work,

Table 5.4: Software cost attributes related to "Project Personnel"

Cost attributes	Reference
Language	[27, 91, 145, 169, 171–173]
Cultural differences	[27, 91, 145, 169, 171–173]
Communication	[27, 91, 169, 171–173]
Trust	[91, 169, 171–173]
Team structure	[91, 171–173]
Team size	[91, 145, 169, 170]
Personnel experience	[25, 27, 145, 172]
Development productivity	[25, 91, 145, 169]
personnel relationships	[25, 27, 171]

Another strongly cited cost attributes in this category is related to communication among teams working. Communication costs arise when there are communication related issues between the sender and recipient sites. Principally these are related to difficulties arising due to telephonic and electronic communication, instead of face-to-face communication [155].

5.4.4 Cost attributes within "Development Environment"

Twelve cost attributes have been identified in the "Development Environment" category. It is by far the category holding the largest number of cost attributes. They are presented in Table 5.5.

TABLE 5.5: Software cost attributes related to "Development Environment"

Cost attributes	Reference
Design and Technology Newness	[25, 30, 91, 145, 169, 171, 172]
Time zone	[27, 91, 145, 169, 171–173]
Process model	[91, 145, 169]
Response delay	[91, 173]
Client involvement	[91, 145, 169]
Project effort	[30, 91, 145, 169, 170, 172]
Project management effort	[25, 30, 91, 145, 169, 172]
Task allocation	[155, 169, 172]
Work pressure	[27, 169, 172]
Work dispersion	[145, 169, 172]
Requirement volatility	[25, 27, 171]
Travel cost	[30]

The most frequently mentioned cost attribute is "Design and Technology Newness". Lack of knowledge on the required new technologies and lack of experience on the used effort estimation technique lead to wrong assumptions, which affect the accuracy of the cost estimates. Another relevant cost attribute presented by the study was the "Time zone" that in many cases makes harder the process of hand off between sites, which also leads to wrong assumptions, compromizing the accuracy of the effort estimates.

5.5 Threats to Validity

5.5.1 Construct Validity

Since the results of this study are drawn from an SLR, the search string used and the libraries queried can be identified as a threat to the construct validity. To minimize these threats we used the "PICO" method to extract initial search terms from the RQ and added synonyms drawn from similar research. We limited our SLR study to four research publication sources that are most common in software engineering (i.e., IEEE Xplore, ScienceDirect, ACM and SpringerLink). However, there are other related research sources that we did not consider in our study that may include relevant studies.

5.5.2 Internal Validity

Internal validity deals with data extraction and analysis. When conducting the SLR, we excluded no articles on the basis of their quality. Although some researchers might prefer to exclude articles of poor quality, including them served to clarify and develop the results of our SLR and allowed us to enrich our finding.

5.5.3 Conclusion Validity

Bias in the selection of software cost attributes in GSD context and data analysis may therefore affect the interpretation of the results. To mitigate this threat, every step in the selection, extraction and analysis of data was validated by means of the systematic process and periodic reviews carried out by the researchers involved in this work.

5.5.4 External Validity

It is unknown if the classification scheme used reflects the actual situation in industry. Interviews or surveys of industry practitioners involved in GSD projects to validate results were not conducted and represents one of the limitations of this study.

5.6 Conclusions

In this chapter, we proposed an up to date framework encompassing 30 cost attributes inherent to GSD projects derived from an SLR comprising 12 primary studies. The framework is divided into three categories based on the SEI taxonomy. The framework is intended for project managers engaging in the development of distributed software projects. However, this framework can only serve as a base for software cost area and must be adapted for each project. In the next chapter, we assess the relevancy of the items presented in the framework using a survey of researchers and industry practitioners.

Chapter 6

Evaluating the proposed Software cost attributes framework for Global Software Development

6.1 Introduction

Developing software in highly distributed environment is becoming a growing trend in the software industry [174]. Global distribution of software projects impacts mainly the software product quality and the software project management process. Managing software in distributed environments comes with great challenges [175]. Cost estimation is one of these challenges that are critical for global software development (GSD) project success [176]. Interest in software cost estimation in GSD has increased in recent years [169]. To better address software cost estimation, software cost attributes (CAs) should be determined. It is now acknowledged that a good software cost estimation that meets distributed project requirements involves the determination of software CAs in early stages of software development.

There is currently a lack of understanding of the role of software CAs in GSD projects in industrial practice [177]. To deal with GSD challenges, a good understanding of how Software CAs interfere on GSD is required. Software CAs are used for the assessment and identification of appropriate project parameters that are inevitable for each model as a preprocessing step. Carrying out relevant project attributes that will guide cost models to better effort approximations could also lead to reduced time and effort required by project managers [176]. According to several studies [169, 178], no thorough empirical studies on the role of software CAs in GSD context has been conducted. In this paper, we therefore elaborate a survey to define the position of software CAs in the design of GSD projects.

This aims of this chapter is to update the results of the entire cost estimation process in distributed projects, by considering the role of software CAs in GSD. To reach this goal, we have collected, analyzed and discussed information from participants with practical experience in software engineering [66]. Software engineering researchers and practitioners can offer a significant knowledge about how CAs are addressed during GSD projects. Therefore, this survey identifies the impact of software CAs on the GSD projects and the manner researchers and practitioners with practical experience use to deal with software CAs [179]. The main target audience of this survey are researchers interested on how the software CAs are handled in GSD projects. We target also practitioners who are interested in discovering which software CAs are important compared to others and how these attributes can be exploited [180].

This chapter is structured as follows: the related work is presented in Sect. 6.2. Sect. 6.3 presents the research method and the details research questions. Sect. 6.4 introduces the results of our study. Sect. 6.5 discusses the results, while Sect. 6.6 presents the paper limitations. Conclusions are presented in Sect. 6.7.

6.2 Related work

In spite of the fact that software cost topic has been considered and discussed for a long time, we have discovered few literature reviews and case studies concerning CAs for GSD [169]. In this section, we present the most relevant work about CAs in literature. Keil et al. [25] have studied the extra cost drivers of global development and have analyzed the importance of each of factors, such as knowledge transfer and effort expended in team building, as a supporter to the general cost of a software development project. They have concluded that project factors should be analyzed to gain insights into comparing development costs for GSD projects as compared to collocated projects.

Peixoto et al. [112] have presented preliminary results from a survey to understand effort estimation in GSD projects. Results have shown that the teams involved in the survey do not have clear criteria to control the choice of an effort estimation technique for a specific project. Moreover, they have found that they can not correlate the cost and time estimation with the number of locations required in the distributed project. Raymond [181] has focused on the estimation models for distributed software development projects. Lamersdorf et al. [27] have developed a model for estimating the cost overhead in GSD projects in view of a set of influencing factors. However, their study represents only one particular environment and can not be easily summed up to different organizations and contexts. After analyzing existing research and studies on software CAs in GSD, we deduce that there is a need to identify the impact of software CAs in GSD context. Our work is thus a beginning stage that contributes towards determining current software CAs in distributed environment.

6.3 Research methodology

A "survey" is a systematic method for gathering information from entities for the purposes of constructing quantitative descriptors of the larger population attributes of which the entities are members [182]. To ensure rigor of our study, and to reduce researcher bias when conducting the survey [183], we have designed a survey protocol. The study follows the five distinct steps of the survey process [184]. Moreover, the paper itself follows the study [66] presenting a detailed structure of a survey in software engineering: Design survey process, develop questions, test & train, collect Data, and finally analyse Data.

6.3.1 Research questions

To better match the goal of our study, we extracted from the questionnaire:

(https://goo.gl/forms/SO9NMmuZXf2vnDL62) two main research questions [141], each one was divided into two sub-questions as shown in Table 6.1.

Table 6.1: Research questions

ID	Question
RQ1	How important are software CAs?
RQ1.1	How important are software CAs in the context of global software projects?
RQ1.2	Are some software CAs more important in distributed project than others?
RQ2	How are software CAs treated in global software projects context?
RQ2.1	What types of outcomes are used to treat software CAs?
RQ2.2	What is the impact of these outcomes on different software CAs?

6.3.2 Survey design

The survey is an appropriate means of gathering information. Three survey types exist [66]: Descriptive, Explanatory and Exploratory surveys. In this study, we have chosen to conduct a descriptive survey to study how software CAs are treated during GSD projects [185]. We have targeted the general community of software engineering practitioners, as well as researchers with practical experience and knowledge about software project management for distributed project. The duration of practical experience was not a restriction factor as long as participants had experience from real projects.

To select relevant participants, we have published the survey on a mailing list encompassing computer engineers. We have also advertised the survey to certain ICGSE conference and workshop participants. Finally, we checked the interest with our personal contacts who, in their turn, spread the survey to other individuals that might be interested in participating in the survey.

6.3.3 Questionnaire description

We have used either interviews or paper based questionnaires depending on the category of the participant. We have used an electronic questionnaire for the rest of participants. In this survey, we have collected informations about the location, practical experience, distribution, educational background, the role of participants, and also the size of participants' organization [186].

Each participant has to choose and specify one project and then answer the questionnaire based on his/her experience on that specific project. Answers can be biased by the perception of each participant depending on his/her evolvement in the project. However, since participants are well targeted, we believe that they have actively participated in the referent project, which make this risk minim. Some questions were optional and others were obligatory [187]. For multiple choice questions, participants chose the answer that suits them best. There are also free text questions. And in most questions, participants can provide additional comments to complete their thoughts. The survey contains different types of questions [66].

6.3.4 Data analysis

To analyze the data, descriptive statistics and qualitative analysis were conducted [188]. Specifically, we have analyzed the variables using correlation analysis and frequency analysis with Fisher's exact test [189] and cross-tabulation. Free text answers were analyzed and coded by all authors.

6.4 Results

6.4.1 Demographic data of respondents

Thirty participants, who have participated in the survey, belong to different organizations and have worked in different projects. The participants were from different countries all over the world.

The countries with the higher participation were Morocco (50%), Spain (33,3%), New Zealand (6,6%), Germany (6,6%), and France (3,3%). Participants were from 3 different continents: Africa (50%), Europe (43,3%), and Australia (6,6%).

6.4.1.1 Participants background

53.3% of participants were engineers (e.g., industrial consultants), while researchers represent 20% of the participants as shown in Table 6.2. Researchers with practical experience had, on average, 3.27 years on research related to GSD. Practitioners had, on average, 4.24 years of experience in IT outsourcing industry.

TABLE 6.2: Distribution of researchers and practitioners with practical experience

Researcher	Practical experience	
	No	Yes
No	0 (0%)	8 (26.7%)
Yes	6 (20%)	16 (53.3%)

Academic background of participants was principally a degree in Computer Science (Bachelor, Master or Engineering degree) for practitioners and a PhD for researchers (see Fig.6.1). All participants had received training in computer science and software engineering.

6.4.1.2 Role of participants

The majority of the 24 practitioners (also practitioners that were at the same time researchers) were developers or project managers (see Fig.6.2), also there were software analysts, and participants with other responsibilities (e.g., architects and designers).

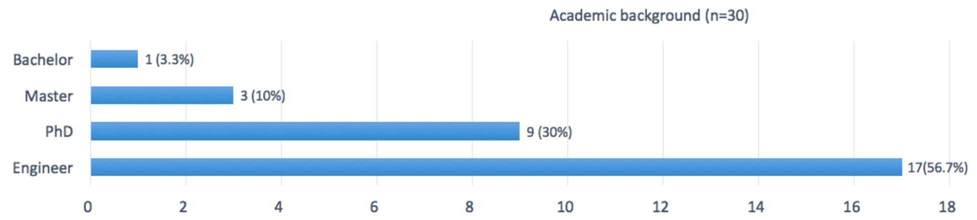

FIGURE 6.1: Academic background

Practitioners, who are neither developers nor project managers in their main role, had based their answers on project they took management responsibilities for.

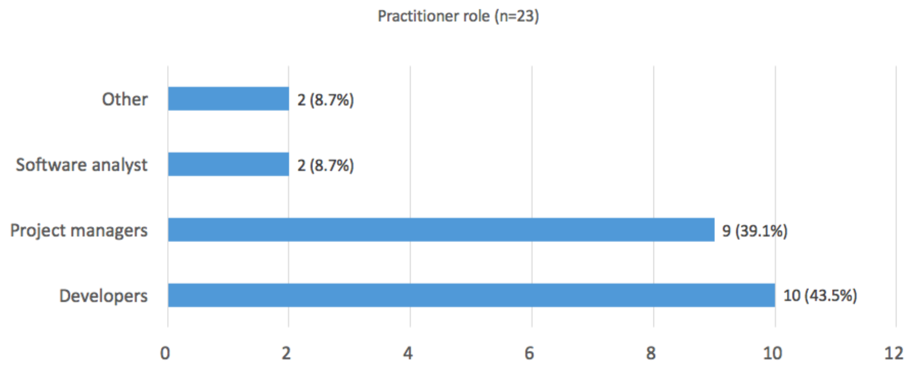

FIGURE 6.2: Role of practitioners

6.4.1.3 Size and number of sites around the world of participants' organization

The majority of answers are provided from participants working in large organizations as shown in Fig.6.3. This figure includes only responses from the 23 practitioners as one practitioner did not give a valid response. Fig.6.3 shows that all practitioners belong to entities with at least 2 sites around the world.

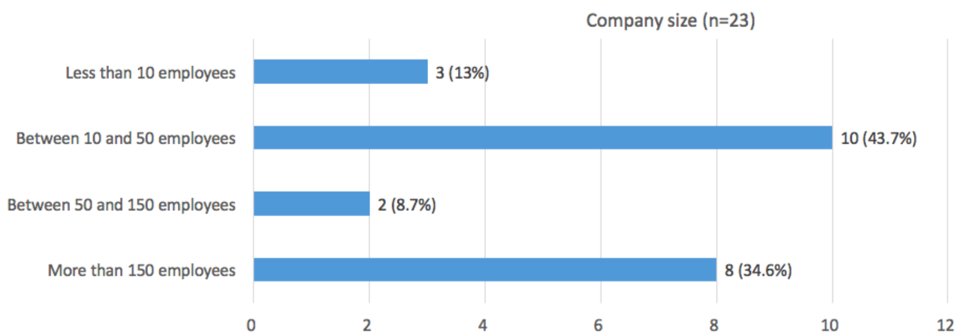

FIGURE 6.3: Company size

6.4.1.4 Context of projects

Participants fill out a brief description of the project they have chosen in their answers. Participants offered a high diversity of projects, e.g., software engineering, telecommunication, e-Government and finance. The quality of responses in many cases did not enable a proper interpretation. For this reason, we have considered all details that concern each project in the analysis of this question's results. Fig.6.4 shows the frequency distribution of project domains (participants' answers were based on their experiences in one specific project). The category "Other" includes domains such as transportation and e-Government.

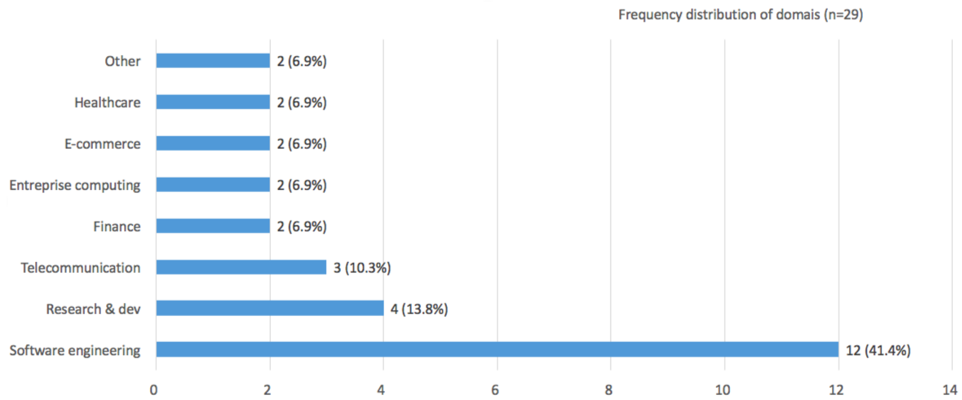

FIGURE 6.4: Frequency distribution of domains

6.4.2 RQ1: How important are software CAs?

6.4.2.1 RQ1.1: How important are software CAs in the context of global software projects?

Time and resources [190] of the system development are the most important requirements besides cost in GSD projects. The research question RQ1.1 compares how CAs are treated compared to both, time and resources. The results are presented in Fig.6.5 (three respondents did not give an answer). The majority of participants indicated that CAs and resources/ time development were considered equally important in a global context. Furthermore, in most projects, CAs were made implicit (63.0%), but still a significant amount of respondents stated that CAs were treated explicitly (37.0%).

To study the dependency between the importance of CAs and the implicit or explicit nature of CAs, we created a cross-tabulation (see Table 6.3). Fisher's exact test led to P=0.28 (superior of 0,05) which signifies that there is no statistically significant relationship between the importance of CAs and their implicit or explicit nature. The result that CAs are implicit shows that special attention is paid to CAs because they represent a major challenge.

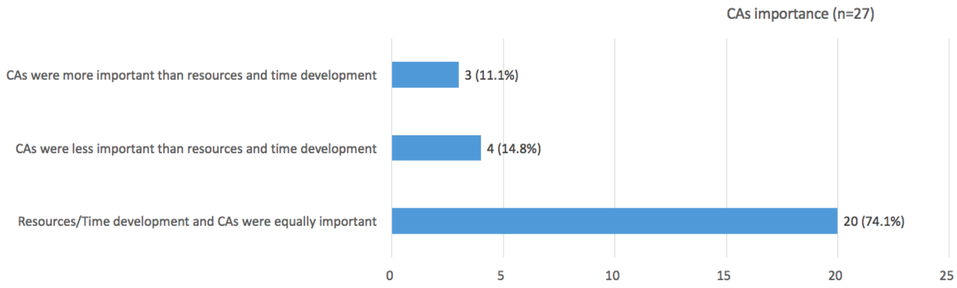

FIGURE 6.5: Role of software CAs

TABLE 6.3: Cross-tabulation of the importance of software CAs and their implicit or explicit nature

	CA explicit	CA implicit	Total
Resources/Time development and CAs were equally important	9 (33.3%)	11 (40.8%)	20 (74.1%)
CAs were less important than resources and time development	1 (3.7%)	3 (11.1%)	4 (14.8%)
CAs were more important than resources and time development	0 (0.0%)	3 (11.1%)	3 (11.1%)
Total	10 (37.0%)	17 (63.0%)	27 (100%)

6.4.2.2 RQ1.2: Are some software CAs more important in distributed project than others?

The attributes selected and used in the questionnaire are extracted from the attributes of ISBSG dataset [19] that includes a vast number of project attributes related to the application domain, programming language used, language type, development technique, resource level, functional size of the software produced among others [185]. We have mapped all software CAs stated by participants. This was done through content analysis of CAs scenarios.

Fig.6.6 presents the frequency distribution of software CAs in distributed project. All participants did specify a valid software CA. In Fig.6.6, we have grouped data related software CAs. Fig.6.6 shows that team size, development methodologies, design and technology newness are the most frequently addressed software CAs.

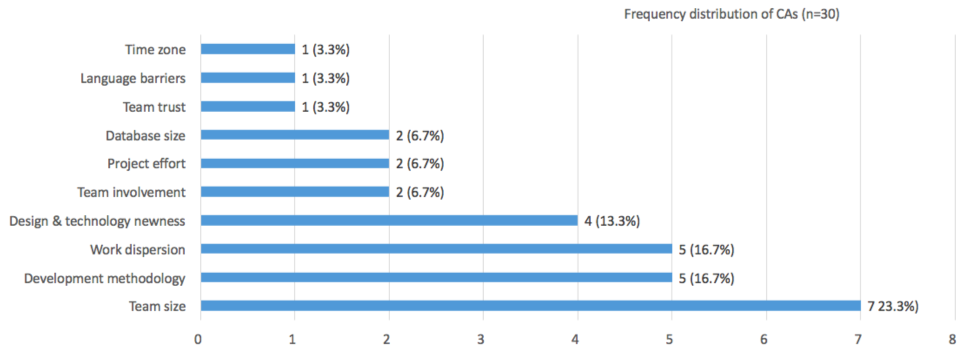

FIGURE 6.6: Frequency distribution of software CAs

6.4.3 RQ2: How are software CAs treated in global software projects context?

6.4.3.1 RQ2.1: What types of outcomes are used to treat software CAs?

We have asked the participants about the most important benefit of using CAs on GSD project. CAs in GSD project allow companies to save cost, remain focused on building internal core competencies, and solve local IT skills shortage. This research shows that, cost saving remains important in outsourcing decision. The results are shown in Fig.6.7 (three respondents did not provide an answer). The category "Other" includes reasons such as flexibility and time-to-market.

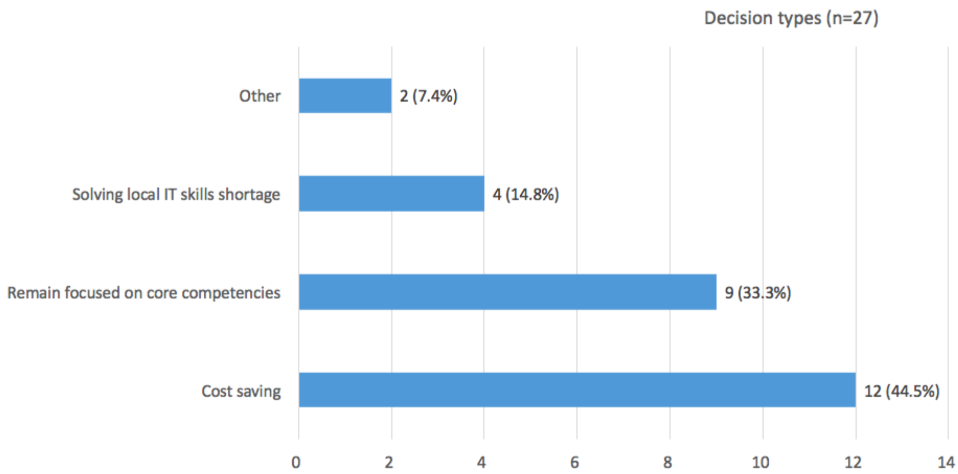

FIGURE 6.7: Decision classification

6.4.3.2 RQ2.2: What is the impact of these outcomes on different software CAs?

We have asked participants what situations had to happen to make these CAs evident or visible to the end users. We obtained 30 responses to this question. Fig.6.8 shows that delays, communication problems and slow performances are the main problems when CAs are no taken into consideration. We have also detected difficulties in establishing a common ground for all stakeholders, and problems on different interpretations of the requirements depending on the location. Two participants stated that they do not consider the CAs in their projects.

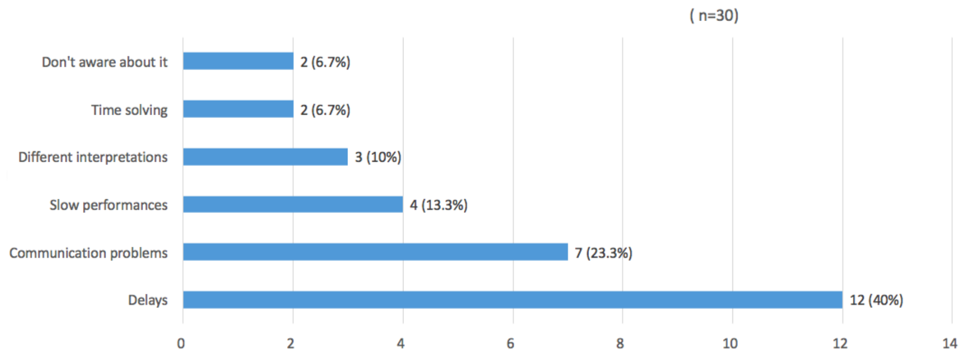

FIGURE 6.8: Cost attributes impact

6.5 Discussion

Cost requirements for GSD projects are the subject of several studies. The larger a software development project is, the higher is the cost required for communication and collaboration between teams. These cost requirements are applicable to any software development projects including GSD projects [25]. Solving local IT skills shortage represents the least significant software cost development requirement, while saving cost represents the main purpose to develop software project globally. Nevertheless, cost-savings in GSD context remain below expectations [191].

Our previous work on GSD [169] contains thoughtful cost estimation requirements and offers attributes to estimate the cost. The results of this survey confirm that the most important CAs are team size and development technologies. The project team size affects cost and schedule decisions, which are also acknowledged as an important factor in project success. Furthermore, development technologies used are important when making decisions about software cost estimation and also the eventual partition of tasks into distributed teams. CAs represent a major challenge for the conception and completion of distributed projects as demonstrated by previous research [22]. Our study does not confirm these results, but it shows that 37% participants indicated that CAs were addressed explicitly. We have shown that CAs are important to address cost requirements and cost factors. The analysis of development methodologies, such as models, is considered a major indicator that helps choose solutions. This consideration could be a driver to consider CAs as management decisions [192].

6.6 Limitations

Failure to evaluate the CAs and their application to estimate cost can bias our study results and lead to erroneous conclusions [65]. Study process includes the determination and involvement of attributes. Controlling the survey sources is a very complicated task to accomplish while designing surveys [68], especially in online questionnaires [66]. Exclusion criteria have been applied to choose relevant answers. Participants with no sufficient experience in software engineering related to GSD are excluded from the study.

Amon the limitations of this study can be the ambiguity of questions. To address this limitation, we have explained the questions several times until our intentions and questions were understood by the selected participants. However, the answers of questions are not only impacted by the understanding of participants, but also the categories of companies and global software projects in which practitioners and researchers worked on. Another limitation can be that participants might answer the questions untruthfully [193]. To solve this problem, we have left the fact of reply voluntary and anonymously. The participants devoted their own time to answer the questionnaire which makes us assume that they have no reason to be inexact [193]. The main limitations are concerned with the issue of generalizing the findings to the software engineering population as our survey involves only a very limited population (30 participants) with GSD projects experience. However, we do believe that our findings apply to a large population that answers the criteria of our survey.

6.7 Conclusions

This empirical study has presented the role of CAs in GSD projects. Using a survey, we have collected data from practitioners involved in IT industry and software engineering projects. We have extended interpretation and discussion of results according to the involvement degree of the researchers and practitioners. For a better judgment, we have measured our results to existing research on software cost field. Our findings not only show how CAs are treated by participants, but also how projects and organizations deal with CAs.

We have noticed that CAs are basically implicit and mainly considered as important as resources and time development of the distributed project. The most important CAs in the GSD context are the team's size and work's dispersion. We have found that CAs are specifically relevant for a GSD project than for a located project. Cost requirement is wrongly treated in practice even more in distributed projects context [194]. Future studies should investigate CAs and their involvement to estimate the software cost in GSD. Moreover, we have noticed that the saving issue is the prime criterion for the majority of decisions taken to supply CAs. Furthermore, we have found that bad estimation was affected mostly by these CAs. Make a saving decision should always take into consideration its influence on CAs.

Chapter 7

Building a software cost estimation taxonomy for GSD projects

7.1 Introduction

Software development effort estimation for Global Software Development (GSD) concerns the prediction of the effort needed to develop a global software project [112]. Development effort is considered to be one of the major components of software costs, particularly as regards global development, and is usually the most challenging effort to predict [27].

To get a more comprehensive understanding of how software cost estimation is practiced by distributed teams, a follow-up study is carried out to elicit the state of the practice on cost estimation in GSD projects [169]. This study identified and aggregated knowledge on cost estimation in GSD from the literature by mean of an SLR and the industry by mean of the survey. The knowledge includes aspects such as approaches used to estimate cost in GSD employed size measures, cost drivers and the context in which estimates are carried out in GSD context. This body of knowledge on cost estimation in GSD must be organized in order to facilitate both future research and practice in this field. A taxonomy is a classification mechanism used in Software Engineering (SE) to organize the body of knowledge [195].

The main purpose of this chapter is to design a software cost estimation taxonomy for GSD projects that will allow managers to rapidly and accurately enact and assess proposed changes. Figure 7.1 illustrates the cost estimation process to build the cost estimate taxonomy.

FIGURE 7.1: Cost estimation process

7.2 Related work

Gumm [196] developed a taxonomy of distribution to classify GSD projects dimensions. In this work the taxonomy is used to provide a basis for discussing the challenges related to GSD projects and was based on an earlier study performed by the same author [197]. The proposed taxonomy uses four different dimensions (physical distribution, organizational distribution, temporal distribution and distribution among stakeholder groups measured on a high- medium- low scale) to classify the distribution of people and artifacts in GSD context.

Laurent et al. [198] proposed a taxonomy and a visual notation to address globally distributed requirements engineering projects. The main goal of the authors was to design the modeling language, including site locations, stakeholder roles, communication flows, critical documents, and supporting tools and repositories. The proposal was based on the findings of eight in-depth interviews with requirements analysts who worked on requirements elicitation, analysis, and specification tasks in globally distributed projects. These Interviews were performed with the team leaders responsible for eliciting and gathering the requirements in each project.

Smite et al. [70] carried out a study specifically focused on evidence of empirical global software engineering. The study proposed a classification scheme to extract data from empirical studies and systematize existing empirical global software engineering studies. The proposed classification scheme helped to categorize the data extracted from the study population, empirical background and results. The study revealed that the collection of papers with empirical data on the subject of the desired systematic review was one of the main challenges, since globally distributed project is at the cutting edge of cross-disciplinary research. The process of deducting and collecting information about empirical background work was another difficulty reported. These highlight the need for thorough descriptions of contexts in which empirical studies are conducted.

More recently, Smite et al. [199] have proposed a taxonomy of sourcing procedures. The taxonomy proposed gives a typical terminology and takes into consideration the classification of GSD projects with spotlight on the sourcing strategies (e.g., Offshore outsourcing, offshore insourcing). The result provided a systematically accumulated set of terms categorized in the form of a taxonomy.

The taxonomy proposed by Smite et al. [199] is considered as knowledge classification approach. These taxonomy of sourcing strategies is the most closely related work to ours, hence our decision to use these taxonomy to understand the GSD project setting. The base of these taxonomy is more exhaustive, giving a more extensive scope of relevant dimensions and clear criteria to classify GSD projects. Furthermore, this taxonomy was also developed with the participation of several GSD experts, which gives the taxonomy more credibility and validity. Therefore, our goal is to propose a taxonomy to classify all dimensions of cost estimation for GSD, and include categories related to empirical focus, subjects of investigation and sources of data collection.

7.3 Research methodology

This section describes the research question and methodology used to design and evaluate the proposed taxonomy.

7.3.1 Research questions

This chapter addresses one Research Question (RQ), which is presented below:

RQ: How to organize the knowledge on cost estimation for GSD projects?

The RQ is answered by organizing the cost estimation knowledge for GSD projects as a taxonomy.

7.3.2 Taxonomy design

The focus of this subsection is to present the method to design the software cost estimation taxonomy used in this study. Usman et al. [200] present a revised and updated method on taxonomies in the field of software engineering. As shown in Table 7.1, these method consists of four phases and thirteen activities .

Table 7.1: Taxonomy design method

Phase	Activity
Planning	A1. Define SE knowledge area A2. Describe the objectives of the taxonomy A3. Describe the subject matter to be classified A4. Select classification structure type A5. Select classification procedure type A6. Identify the sources of information
Identification and extraction	A7. Extract all terms A8. Perform terminology control
Design and construction	A9. Identify and describe taxonomy dimensions A10. Identify and describe categories of each dimension A11. Identify and describe the relationships A12. Define the guidelines for using and updating the taxonomy
Validation	A13. Validate the taxonomy

Phase 1: Planning

Planning represents the first phase wherein basic decisions about the taxonomy implementation and design are made. In this phase, six activities are defined as shown in Table 7.1.

In activity **A1**, the SE knowledge area is selected and described to make easier the understanding of the context of the taxonomy and thus there application. The taxonomy proposed in this chapter is about cost estimation for GSD context. Cost estimation plays an important role in managing distributed projects during release and planning. Cost estimation falls within the scope of the ”Software engineering management” knowledge area in SWEBOK version 3 [44].

In activity **A2**, the main objectives and scope of the taxonomy is to propose a classification scheme that can be used to characterize cost estimation activities for GSD projects. A number of studies, included in the SMS on software cost estimation for GSD in Chapter 4, have not reported important information related to the context, techniques and predictors used during cost estimation. therefore, the proposed taxonomy could be used by researchers and practitioners to consistently report and remind important aspects related to software cost estimation for GSD projects.

In activity **A3**, the subject matter for the classification defines what exactly is classified in the taxonomy. Cost estimation activities of the projects that are globally developed is the subject matter of this taxonomy.

In activity **A4**, an appropriate classification structure type is selected. Four basic classification structures are defined: hierarchy, tree, paradigm and faceted analysis [201]. To structure our taxonomy, faceted classification is selected, since it is suitable for evolving areas, such as software cost estimation for GSD. In faceted classification-based taxonomies, the subject matter is classified from multiple perspectives (facets). Each facet is independent and has its own attributes, making the facet-based taxonomies easily evolvable [201].

In activity **A5**, an appropriate classification procedure is determined. These type can be qualitative, quantitative or both. Each facet of our taxonomy has a set of possible values. Based on extracted data, the qualitative procedure is used to select relevant facet values to characterize a specific estimation activity. In some cases, to assign a value is impossible simply due to insufficient data.

In activity **A6**, the data sources and data collection methods are identified to facilitate the prospection of knowledge related to the subject matter and taxonomy. These data sources are selected from peer-reviewed empirical studies on cost estimation for GSD published in literature.

Phase 2: Identification and extraction

In this phase the relevant data required by the organization is identified and extracted. Two activities are defined as shown in Table 7.1.

In activity **A7**, the terms and concepts relevant to the taxonomy are extracted from the sources identified in the first phase “planning”.

In activity **A8**, inconsistencies in the extracted data are identified and removed.

Phase 3: Design and construction

In this phase, the taxonomy is designed and constructed by mean to identify dimensions and categories therein along which the extracted data items could be organized. Four activities are defined as shown in Table 7.1.

In activity **A9**, the taxonomy dimensions are identified and described. They represent the main dimensions or perspectives under which subject matter entities are classified. Facet-based classification taxonomy structure must have multiple dimensions (at least two dimensions).

In activity **A10**, the categories for each of the dimensions are identified and described, each dimension must have at least two categories.

In activity **A11**, the relationships between dimensions and categories are identified and described. Note that in some cases there is no relationship between dimensions, i.e. this activity might be skipped.

In activity **A12**, the guidelines are provided to facilitate the adoption and evolution of the taxonomy.

Phase 4: Validation

Validation represents the last phase of this method to ensure that the designed taxonomy useful for users to achieve their goals. In this phase, only one activity is defined as shown in Table 7.1.

In activity **A13**, the taxonomy can be validated through benchmarking. Since there is no existing taxonomy on software cost estimation for GSD, benchmarking our taxonomy against existing ones is impossible.

7.4 Results

In this section, we present the software cost estimation taxonomy to answer the proposed RQ.

7.4.1 RQ: Organizing the knowledge on cost estimation in GSD

Taxonomies represent an effective tool to organize and communicate the knowledge in an area [195]. We have organized the identified knowledge on cost estimation for GSD as a taxonomy. The proposed taxonomy was developed according to the method presented in Section 7.3. In this subsection, we describe the results of this method, i.e. the software cost estimation taxonomy.

Four dimensions are extracted and placed at the top level of the taxonomy, as shown in Figure 7.2. A dimension consists of all facets that are interrelated. These four dimensions characterize the first level of taxonomy, and give an outline of the taxonomy at a more extensive level.

The estimation context represents the collection of those facets that define and characterize the context in which the estimation activity in a distributed project is carried out.

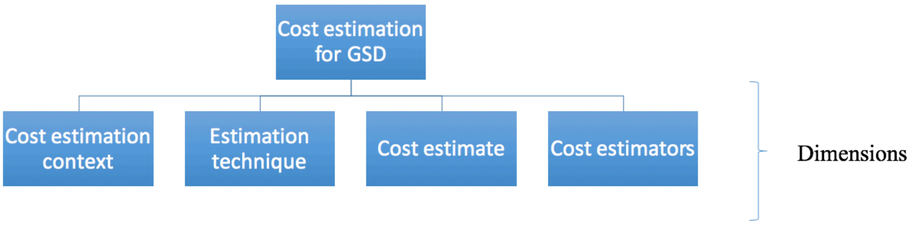

FIGURE 7.2: First level of cost estimation taxonomy for GSD

So as to fully characterize a specific software cost estimation activity of GSD project, facets of all dimensions should to be described. Each of these four dimensions and their facets are described in detail below.

7.4.1.1 Cost estimation context

The context is a central concept in empirical software engineering. It is one of the distinctive features of the discipline and it is an inseparable part of software practice. Context refers to a broad perspective, and it needs to be properly captured, reported , and contextualized in the empirical SE studies to communicate the applicability of the research findings. Thus, context draws attention to what resources are nearby, and when and where to use the reported findings [202], [203].

Seven facets are extracted from context dimension. These facets and their possible values are presented in Figure 7.3. They are described below:

- *Planning:* Estimation supports planning at various levels in GSD context. This mainly includes release and sprint planning [83], while some teams may also make estimates during daily meetings. Project bidding is another level at which companies must estimate the total development cost in advance to bid for the projects.
- *Project activities:* This facet describes which development activities are accounted for in software cost estimation. For example, The product life cycle describes maintenance activity, or the total cost estimate do not include the time spent on maintenance.
- *Project domain:* This facet represents the domain of project for which the software cost development is being estimated. Different domains could lead to different sets of cost estimation. We have used the categories from the project domains reviewed in Chapter 6.
- *Project setting:* This facet represents the setting in which the global software teams are developing the project. Smite et al. [199] proposed a global software

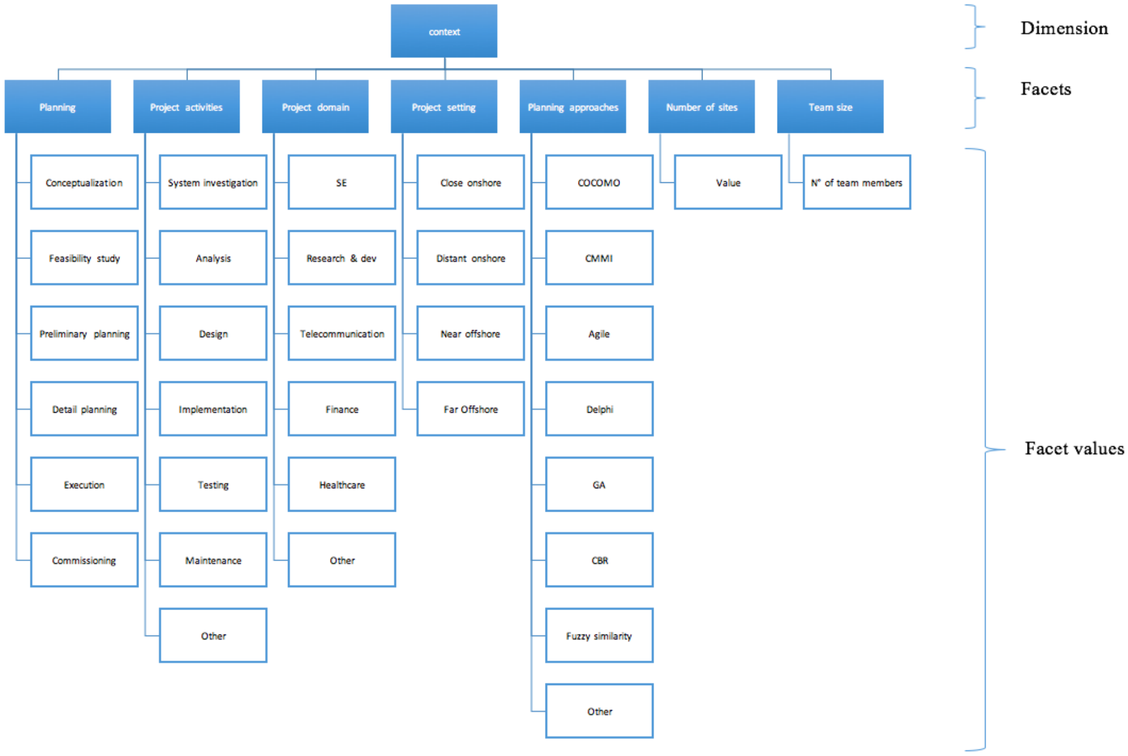

FIGURE 7.3: Cost estimation context dimension

engineering taxonomy that characterizes two broad settings of the global teams: onshore and offshore.

- *Planning approaches:* This facet documents the planning approach practiced by GSD team.
- *Number of sites:* It records the number of sites for the GSD project, and thus conveys important information related to the different site across the world.
- *Team size:* It documents the team size which is responsible for developing the estimated tasks.

7.4.1.2 Estimation technique

This dimension includes the facets that are related to estimation techniques. those facets should be reported to characterize a GSD team's estimation activity. Figure 7.4 describes the facets of this dimension and their corresponding values.

- *Estimation technique:* This documents the estimation techniques applied for GSD projects. According to the SMS results presented in Chapter 4, the cost estimation techniques for GSD projects are expert judgment, machine learning and non-machine learning.

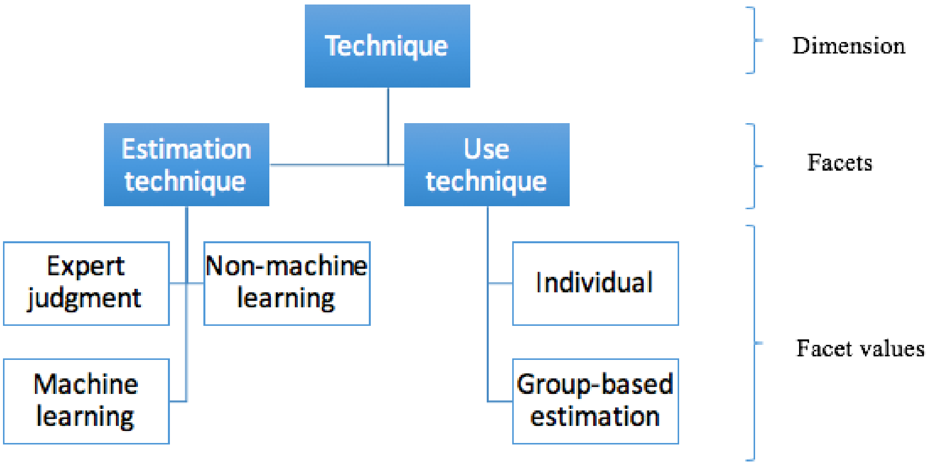

FIGURE 7.4: Estimation technique dimension

- *Use technique:* In GSD, there are different types of cost estimation techniques. An individual or a group of experts can use these techniques. This facet documents whether the effort was estimated using an estimation technique based on individuals or groups.

7.4.1.3 Cost estimate

The main output of the estimation activity is cost estimates. The facets proposed in this dimension define cost estimate. Those facets with their corresponding values are presented in Figure 7.5.

- *Estimated cost:* This facet documents the estimated cost that represents the main output of the estimation activity.
- *Actual cost:* It is important to have the actual cost at the end of planning, to enable comparison with the estimated cost.
- *Estimation dimension:* This facet documents the important and critical dimensions of estimation, e.g. estimation of development effort as total effort hours.
- *Accuracy measure:* This facet records cost performance ways to assess the accuracy of the applied estimation technique.

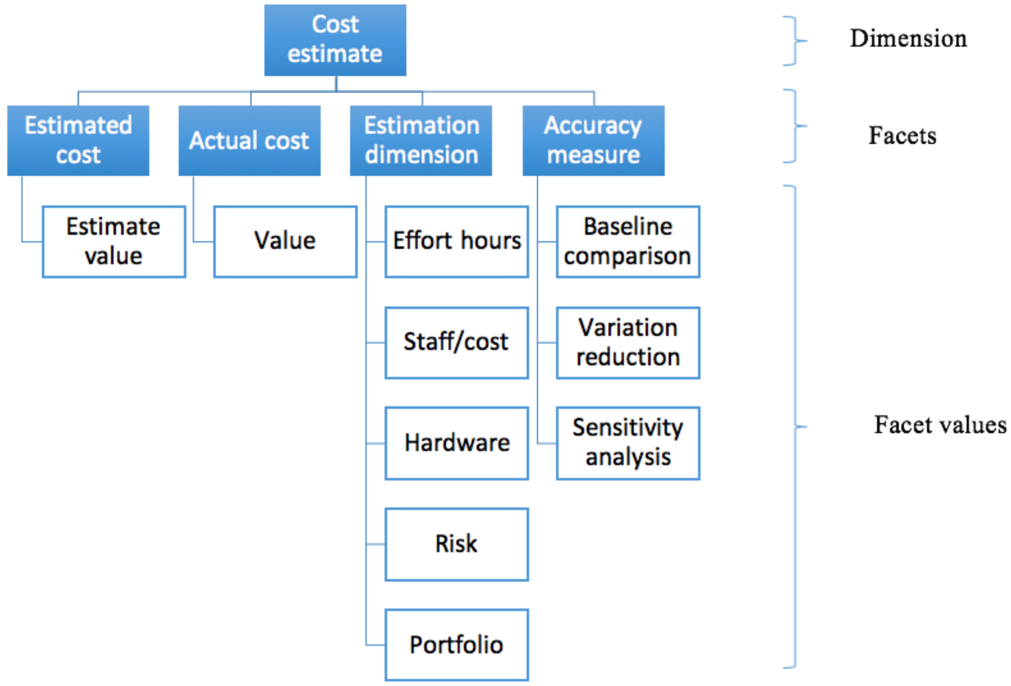

Figure 7.5: Cost estimate dimension

7.4.1.4 Cost estimators

Cost estimators play an important role in calculating costs. They consist of cost drivers such as size, team capabilities, product requirements, etc. One of the most required development cost is related to project size. Five facets are collected regarding the cost estimators dimension. These facets and their possible values are presented in Figure 7.6.

- *Product size:* In general, the development cost is strongly correlated with product size [204]. This facet documents whether distributed teams use size as an estimator and which statistical analysis is used to represent this size.
- *Team experience:* A development team experiences with global software development projects impact the required cost [205]. This facet describes whether a team experience was considered or not in arriving at the cost estimates.
- *Team structure:* Distribution of skills and team structure impact the required effort [206]. This facet documents whether the structure of the team members was considered or not during the cost estimation session.
- *Product requirement:* Strict product requirements increase the development cost [207]. This facet records which product requirements were considered in arriving at cost estimates.

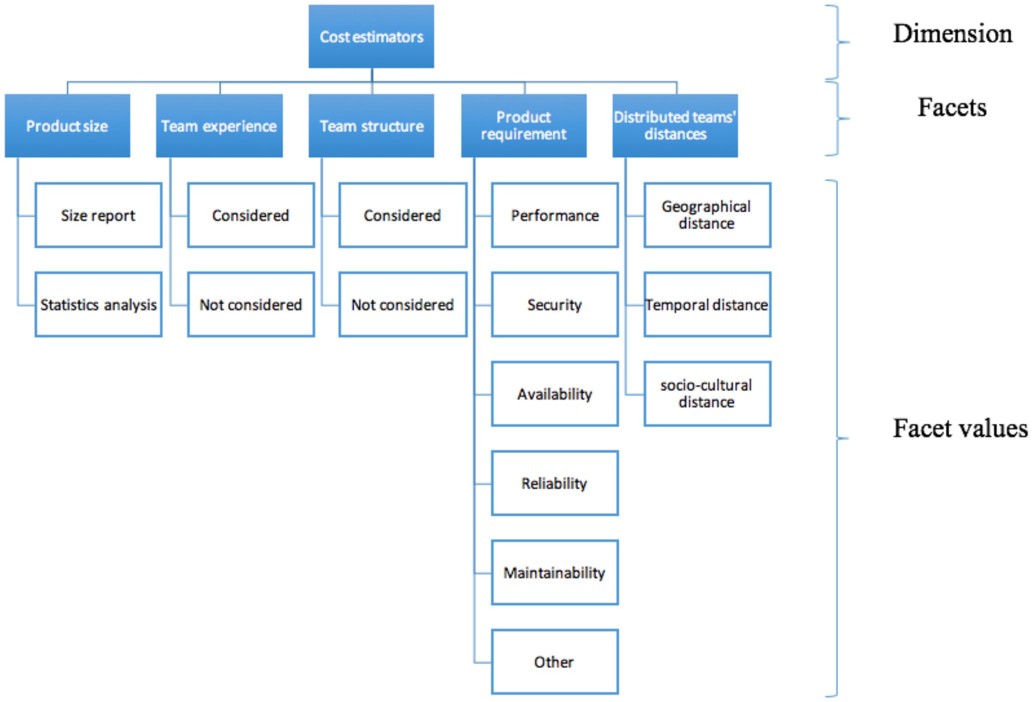

FIGURE 7.6: Cost estimators dimension

- *Distributed teams' distances:* The geographical, temporal, and socio-cultural distances between global development teams increase the development cost due to the increased complexity of the collaboration and communication [4].

7.5 Threats to validity

The results of the proposed taxonomy may have been influenced by the coverage of the study search, and also the inaccuracy in study data extraction. Four types of validity threats [156] of the study results are therefore discussed in the following subsections.

Construct validity is concerned with issues that may arise due to poor gathering and recording of data, also, exactitude of the interpretation of the concepts studied and the completeness of the relevant studies collected. In this study, the data identified and aggregated are extracted from the previous SLR (Chapter 4) and survey (Chapter 6). The extracted data was used as main input to the taxonomy designed in this study. To ensure the correct interpretation of these data, we checked the contributions of the concepts in related literature and all the authors discussed these data in order to reach a consensus as to their use and contribution.

Internal validity is concerned with analyzing extracted data. Threats to internal validity are important as our study does not present an evaluation of the taxonomy. In future studies, we plan to evaluate and assess the taxonomy by using it to characterize estimation cases from the literature and the industry.

Conclusion validity is to ensure that reasonable conclusions are drawn on the basis of data collected and that problems such as the bias of researchers do not lead to incorrect conclusions. We used a taxonomy design method to systematically organize the knowledge on software cost estimation for GSD as taxonomy in a systematic manner.

External validity is concerned with the representativeness of the selected studies as regard the overall goal of the study. The results of this study were considered with regard to the cost estimation for GSD projects. These results can serve as a starting point for researchers and practitioners working in this field to further improve the completeness and usefulness of the proposed taxonomy.

7.6 Conclusion

The development of taxonomies helps to structure, generalize and share existing knowledge and to advance research [195]. We have organized the existing body of knowledge on cost estimation in GSD as a taxonomy. The taxonomy has been systematically developed by following a taxonomy design method. One research question was addressed by incorporating five dimensions to organize knowledge on cost estimation for GSD projects. The main usage for our taxonomy is to provide a basis for researchers to classify their own studies and related studies on cost estimation for GSD field. These taxonomy could therefore be used as a tool to develop a cost estimation knowledge repository to better understand and improve the cost estimation practice in the global development context in the long term.

The usefulness of the taxonomy has not been demonstrated in the study. We plan to apply the developed taxonomy on data extracted and reported in the literature to characterize cost estimation cases.

Chapter 8

Conclusion and perspective

This chapter analyses the work presented and its limitations and summarizes the research findings. Section 8.1 details the research carried out during this thesis, section 8.2 presents the contributions of this thesis to the fields of GSD, SPM and SCE and section 8.3 concludes this thesis by pointing out future areas of research.

8.1 Summary of carried out research

In Chapter 3, we reported an SMS that explored what approaches support SPM activities in GSD. The study provided first an overview of the research area and identified 84 relevant studies related to SPM approaches used in GSD. The study identified the publication source and frequency of relevant literature and determined its research type and approach. The selected studies provided a list of SPM activities for GSD projects which were classified according several knowledge area. A list of benefits and limits of the use of approaches in SPM for GSD were classified according each type approach. The study determined a lack of approaches that support communication, coordination, collaboration and performance in GSD projects.

In Chapter 4, we conducted an SMS of cost estimation in the context of GSD projects. The study served both research and practice. The SMS identified a wide spectrum of software estimation techniques, activities and cost drivers for GSD projects. The selected studies provided software cost contribution as regards cultural, language and time zone differences, which are directly related to making the achievement of globally performed software projects.

In Chapter 5, we conducted an SLR to identify software cost attributes for GSD available in literature. We applied a refined search strategy by (i) expanding the search terms used, (ii) including digital libraries to be queried and (iii) using quality assessment as a study selection criteria. The list of identified cost attributes provides an initial validation of the results. The proposed framework was divided into three categories based on the SEI taxonomy.

In Chapter 6, we assessed the relevancy of the items composing the framework proposed in Chapter 5 using a questionnaire based survey. The results of the survey ranked the importance of each according to practitioners while interpreting results differences between project managers and developers. The framework still can be further ameliorated through iterative surveys and feedback from case studies.

In Chapter 7, we built a taxonomy to organize the knowledge on software cost estimation for GSD projects. The proposed taxonomy offered a faceted classification scheme to characterize cost estimation activities of distributed projects.

8.2 Contributions of the thesis

In a GSD context, this work investigated the state of research in the area of SPM and more specifically software cost estimation activities. A framework to identify software cost attributes in a GSD context was subsequently proposed. The framework was populated through an SLR and validated by researchers and industry practitioners through a questionnaire - based survey. This thesis has made the following contributions to the fields of GSD, SPM and SCE:

1. **A summary of research knowledge concerning the usage of approaches when managing GSD projects:**
   The emergence of GSD as a new software development paradigm has prompted many researchers to investigate the use of approaches in GSD context management. Researchers analyzed existing approaches to investigate their impact on coordination and collaboration. By using a systematic study and a carefully crafted search strategy, we were able to identify a significant number of relevant publications ranging from the year 1998 to the year 2015 and report on this research area's landscape. On one hand, the information provided can be help academicians wishing to expand their research to GSD by pointing out relevant venues for publication and involving collaborative research groups. On the other hand, it can be beneficial to industry practitioners wishing to get involved in GSD activities. It highlights and describes the benefits and limits that can be faced by software

companies. In particular, project managers are advised to consider the list of approaches and their characteristics when selecting the approaches to be used when carrying out SPM activities.

2. **A comprehensive software cost attributes framework intended for GSD context:**
Concerned by the lack of approaches supporting SCE activities in the context of GSD, we intended to develop a framework for this purpose. First, through a systematic approach, we collected software cost attributes included in the literature. Then, we classified the various items into easily manageable categories. Next, we validated the proposed framework using a survey of researchers and practitioners. Finally, we used the resulting software cost attributes framework to construct a SCE taxonomy for GSD context. In our opinion, and to the greatest extent of our knowledge, there are no other taxonomies that use the latest research evidence and are intended by project managers for GSD use. We believe this taxonomy, to be a well welcomed addition to the arsenal knowledge by GSD project managers.

8.3 Future work

The results of this research can be used specifically in relation to software cost estimation as an essential basis for further research in the area surrounding GSD. We identified the following probable future studies:

1. **Assess the effects of software cost attributes:**
While there are definitely some occurrences of empirical evidence of the effect of attributes in software cost estimation, there are almost none that document the effect of the appropriate software cost attributes on GSD context. In order to enhance the contribution of the proposed software cost attributes, it is recommended to keep assessing the list of appropriate attributes using either the evidence provided in relevant literature, by conducting experiments or using reports of case studies. In this regard, the authors intend to keep using empirical data to further enhance the accuracy of software cost estimation approaches.

2. **Extend a software cost estimation taxonomy for GSD:**
The list of dimensions in our taxonomy does not represent an exhaustive list of relevant dimensions related to software cost estimation for GSD. Therefore, we intend to conduct further investigation to identify other dimensions that could be incorporated into the taxonomy, so that GSD projects could be classified in a more comprehensive way. More specifically, we intend to identify dimensions related to

how software cost estimation approaches are framed in GSD projects and how these factors relate to the cost of GSD projects.

List of Publications

Publications in peer-reviewed scientific journals:

- M. El Bajta, A. Idri, J. N. Ros, J. L. Fernández-Alemán, J. M. Carrillo Gea, F. García, and A. Toval. Software Project Management Approaches for Global Software Development: A Systematic Mapping Study. Tsinghua Science and Technology, 29. 2018.
  DOI: 10.26599 / TST.2018.9010029

Chapter book:

- M. El Bajta, A. Idri, J. N. Ros, J. L. Fernández-Alemán, and A. Toval. Estimation of costs and time for the development of distributed software. In Software Project Management for Distributed Computing (pp. 25-42). Springer, Cham, 2017.

Publications in peer-reviewed conference proceedings:

- M. El Bajta, A. Idri, J. L. Fernández-Alemán, J. N. Ros, and A. Toval. Software cost estimation for global software development a systematic map and review study. In Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 IEEE 10th International Conference on (pp. 197-206), April 2015.
  DOI: 10.5220/0005371501970206
- M. El Bajta. Analogy-based software development effort estimation in global software development. In Global Software Engineering Workshops (ICGSEW), 2015 IEEE 10th International Conference on (pp. 51-54), July 2015.
  DOI 10.1109/ICGSEW.2015.19
- M. El Bajta, A. Idri, J. N. Ros, J. L. Fernández-Alemán, and A. Toval. Software Cost Attributes in Global Software Development Projects: A Survey. In Proceedings of the 9th International Conference on Information Management and

Engineering (pp. 96-101). ACM. Oct 2017.
DOI: https://doi.org/10.1145/3149572.3149607

Publications under review:

- M. El Bajta, and A. Idri. Building a software cost estimation taxonomy for Global Software Development projects. Under revision in a conference.

Bibliography

[1] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, EASE'08, pages 71–80, Bari, Italy, 2008. Blekinge Institute of Technology.

[2] James D Herbsleb and Deependra Moitra. Global software development. *IEEE software*, 18(2):16–20, 2001.

[3] Juyun Cho. Globalization and global software development. *Issues in information systems*, 8(2):287–290, 2007.

[4] Helena Holmstrom, Eoin Ó Conchúir, J Agerfalk, and Brian Fitzgerald. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *Global Software Engineering, 2006. ICGSE'06. International Conference on*, pages 3–11. IEEE, 2006.

[5] Eoin Ó Conchúir, Pär J Ågerfalk, Helena H Olsson, and Brian Fitzgerald. Global software development: where are the benefits? *Communications of the ACM*, 52 (8):127–131, 2009.

[6] Christof Ebert, Marco Kuhrmann, and Rafael Prikladnicki. Global software engineering: Evolution and trends. In *Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on*, pages 144–153. IEEE, 2016.

[7] Srinivas Krishna, Sundeep Sahay, and Geoff Walsham. Managing cross-cultural issues in global software outsourcing. *Communications of the ACM*, 47(4):62–66, 2004.

[8] Lucas Layman, Laurie Williams, Daniela Damian, and Hynek Bures. Essential communication practices for extreme programming in a global software development team. *Information and Software Technology*, 48(9):781–794, 2006.

[9] Gabriela Avram, Liam Bannon, John Bowers, Anne Sheehan, and Daniel K Sullivan. Bridging, patching and keeping the work flowing: Defect resolution in distributed software development. *Computer Supported Cooperative Work (CSCW)*, 18(5-6):477, 2009.

[10] Eoin Ó Conchúir, Helena Holmstrom, J Agerfalk, and Brian Fitzgerald. Exploring the assumed benefits of global software development. In *Global Software Engineering, 2006. ICGSE'06. International Conference on*, pages 159–168. IEEE, 2006.

[11] Muhammad Ilyas and Siffat Ullah Khan. Software integration in global software development: challenges for gsd vendors. *Journal of Software: Evolution and Process*, 29(8):e1875, 2017.

[12] Siri-on Setamanit and David Raffo. Identifying key success factors for globally distributed software development project using simulation: A case study. In *International Conference on Software Process*, pages 320–332. Springer, 2008.

[13] Iso/iec/ieee approved draft international standard - systems and software engineering – software life cycle processes. *ISO/IEC/IEEE P12207-FDIS-1707, June 2017*, pages 1–156, Jan 2017.

[14] Ana Magazinius, Sofia Börjesson, and Robert Feldt. Investigating intentional distortions in software cost estimation–an exploratory study. *Journal of Systems and Software*, 85(8):1770–1781, 2012.

[15] Hareton Leung and Zhang Fan. Software cost estimation. In *Handbook of Software Engineering and Knowledge Engineering: Volume II: Emerging Technologies*, pages 307–324. World Scientific, 2002.

[16] B. A. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. Technical report, Software Engineering Group, Keele University and Department of Computer Science University of Durham, 2007.

[17] David Budgen and Pearl Brereton. Performing systematic literature reviews in software engineering. In *Proceedings of the 28th international conference on Software engineering*, pages 1051–1052. ACM, 2006.

[18] Michael Coughlan, Patricia Cronin, and Frances Ryan. Survey research: Process and limitations. *International Journal of Therapy and Rehabilitation*, 16(1):9–15, 2009.

[19] Marta Fernández-Diego and Fernando González-Ladrón-De-Guevara. Potential and limitations of the isbsg dataset in enhancing software engineering research: A mapping review. *Information and Software Technology*, 56(6):527–544, 2014.

[20] Erran Carmel and Ritu Agarwal. Tactical approaches for alleviating distance in global software development. *IEEE software*, 18(2):22–29, 2001.

[21] Miguel Jiménez, Mario Piattini, and Aurora Vizcaíno. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, 2009:3, 2009.

[22] James D Herbsleb. Global software engineering: The future of socio-technical coordination. In *2007 Future of Software Engineering*, pages 188–198. IEEE Computer Society, 2007.

[23] John Noll, Sarah Beecham, and Ita Richardson. Global software development and collaboration: barriers and solutions. *ACM inroads*, 1(3):66–78, 2010.

[24] James D. Herbsleb and Audris Mockus. An empirical study of speed and communication in globally distributed software development. *Software Engineering, IEEE Transactions*, 29(6):481–494, 2003.

[25] Patrick Keil, Daniel J Paulish, and Raghvinder S Sangwan. Cost estimation for global software development. In *Proceedings of the 2006 international workshop on Economics driven software engineering research*, pages 7–10. ACM, 2006.

[26] Theodore Forbath, Peter Brooks, and Anand Dass. Beyond cost reduction: Using collaboration to increase innovation in global software development projects. In *Proceedings of the 3rd IEEE International Conference on Global Software Engineering (ICGSE)*, pages 205–209, 2008.

[27] Ansgar Lamersdorf, Jurgen Munch, Alicia Fernandez-del Viso Torre, Carlos Rebate Sánchez, and Dieter Rombach. Estimating the effort overhead in global software development. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, pages 267–276. IEEE, 2010.

[28] Mohammad Muhairat, Saleh Aldaajeh, and Rafa E Al-Qutaish. The impact of global software development factors on effort estimation methods. *European Journal of Scientific Research*, 46(2):221–232, 2010.

[29] Mohammad Azzeh. Software cost estimation based on use case points for global software development. In *Computer Science and Information Technology (CSIT), 2013 5th International Conference on*, pages 214–218. IEEE, 2013.

[30] Darja Šmite, Ricardo Britto, and Rini Van Solingen. Calculating the extra costs and the bottom-line hourly cost of offshoring. In *Proceedings of the 12th International Conference on Global Software Engineering*, pages 96–105. IEEE Press, 2017.

[31] D Damian and D Moitra. Introduction: global software development: how far have we come? *IEEE Softw*, 5:17–19.

[32] Pär J Ågerfalk, Brian Fitzgerald, Helena Holmström Olsson, and Eoin Ó Conchúir. Benefits of global software development: the known and unknown. In *International Conference on Software Process*, pages 1–9. Springer, 2008.

[33] Alexander Boden, Gabriela Avram, Liam Bannon, and Volker Wulf. Knowledge management in distributed software development teams-does culture matter? In *Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, pages 18–27. IEEE, 2009.

[34] Ita Richardson, Valentine Casey, John Burton, and Fergal McCaffery. Global software engineering: A software process approach. In *Collaborative Software Engineering*, pages 35–56. Springer, 2010.

[35] Daniela Damian. Workshop on global software development. *SIGSOFT Software. Eng. Notes*, 27(5), 2002.

[36] James D. Herbsleb and Audris Mockus. An empirical study of speed and communication in globally distributed software development. *IEEE Transactions on software engineering*, 29(6):481–494, 2003.

[37] Audris Mockus and James Herbsleb. Challenges of global software development. In *Software Metrics Symposium, 2001. METRICS 2001. Proceedings. Seventh International*, pages 182–184. IEEE, 2001.

[38] Miguel Jiménez, Mario Piattini, and Aurora Vizcaíno. Challenges and improvements in distributed software development: A systematic review. *Advances in Software Engineering*, 2009:3, 2009.

[39] Fabio QB da Silva, Catarina Costa, A Cesar C Franca, and Rafael Prikladinicki. Challenges and solutions in distributed software development project management: A systematic literature review. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, pages 87–96. IEEE, 2010.

[40] Christof Ebert and Philip De Neve. Surviving global software development. *IEEE software*, 18(2):62–69, 2001.

[41] Siri-on Setamanit, Wayne Wakeland, and David Raffo. Planning and improving global software development process using simulation. In *Proceedings of the 2006 international workshop on Global software development for the practitioner*, pages 8–14. ACM, 2006.

[42] PMBOK Guide. A guide to the project management body of knowledge. In *Project Management Institute*, volume 3, 2004.

[43] Ken Lynch. Project management institute (pmi). *The Quality Assurance Journal*, 8(2):126–128, 2004.

[44] Pierre Bourque, Richard E Fairley, et al. *Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0.* IEEE Computer Society Press, 2014.

[45] Iso/iec/ieee approved draft international standard - systems and software engineering - software life cycle processes. pages 1–156, 2017.

[46] Vahid Khatibi and Dayang NA Jawawi. Software cost estimation methods: A review 1. 2011.

[47] NASA NASA. Cost estimating handbook. *Washington, DC*, 2008.

[48] Michael Bramble, Jairus Hihn, John Hackney, Mori Khorrami, and Erik Monson. Handbook for software cost estimation. 2003.

[49] Barry W Boehm et al. *Software engineering economics*, volume 197. Prentice-hall Englewood Cliffs (NJ), 1981.

[50] Tom Gilb and Susannah Finzi. *Principles of software engineering management*, volume 11. Addison-Wesley Reading, MA, 1988.

[51] He Zhang and Muhammad Ali Babar. Systematic reviews in software engineering: An empirical investigation. *Information and Software Technology*, 55(7):1341–1354, 2013.

[52] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*, 80(4): 571–583, 2007.

[53] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, page 38. ACM, 2014.

[54] Kai Petersen, Robert Feldt, Shahid Mujtaba, and Michael Mattsson. Systematic mapping studies in software engineering. In *EASE*, volume 8, pages 68–77, 2008.

[55] Aurora Vizcaíno, Félix García, José Carlos Villar, Mario Piattini, and Javier Portillo. Applying q-methodology to analyse the success factors in gsd. *Information and Software Technology*, 55(7):1200–1211, 2013.

[56] Pearl Brereton, Barbara A Kitchenham, David Budgen, Mark Turner, and Mohamed Khalil. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of systems and software*, 80(4): 571–583, 2007.

[57] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.

[58] Jianfeng Wen, Shixian Li, Zhiyong Lin, Yong Hu, and Changqin Huang. Systematic literature review of machine learning based software development effort estimation models. *Information and Software Technology*, 54(1):41–59, 2012.

[59] Barbara Kitchenham, Rialette Pretorius, David Budgen, O Pearl Brereton, Mark Turner, Mahmood Niazi, and Stephen Linkman. Systematic literature reviews in software engineering–a tertiary study. *Information and Software Technology*, 52 (8):792–805, 2010.

[60] Roel Wieringa and Hans Heerkens. Designing requirements engineering research. In *Comparative Evaluation in Requirements Engineering, 2007. CERE'07. Fifth International Workshop on*, pages 36–48. IEEE, 2007.

[61] Benoît Godin. The linear model of innovation: The historical construction of an analytical framework. *Science, Technology, & Human Values*, 31(6):639–667, 2006.

[62] Francis Lau. Toward a framework for action research in information systems studies. *Information Technology & People*, 12(2):148–176, 1999.

[63] Barbara Kitchenham, Shari Lawrence Pfleeger, and Norman Fenton. Towards a framework for software measurement validation. *IEEE Transactions on software Engineering*, 21(12):929–944, 1995.

[64] Shari Lawrence Pfleeger and Barbara A Kitchenham. Principles of survey research: part 1: turning lemons into lemonade. *ACM SIGSOFT Software Engineering Notes*, 26(6):16–18, 2001.

[65] Barbara Kitchenham and Shari Lawrence Pfleeger. Principles of survey research: part 5: populations and samples. *ACM SIGSOFT Software Engineering Notes*, 27 (5):17–20, 2002.

[66] David Ameller, Matthias Galster, Paris Avgeriou, and Xavier Franch. A survey on quality attributes in service-based systems. *Software Quality Journal*, 24(2): 271–299, 2016.

[67] Kate Kelley, Belinda Clark, Vivienne Brown, and John Sitzia. Good practice in the conduct and reporting of survey research. *International Journal for quality in health care*, 15(3):261–266, 2003.

[68] Marcus Ciolkowski, Oliver Laitenberger, Sira Vegas, and Stefan Biffl. Practical experiences in the design and conduct of surveys in empirical software engineering. In *Empirical methods and studies in software engineering*, pages 104–128. Springer, 2003.

[69] Syed Roohullah Jan, Faheem Dad, Nouman Amin, Abdul Hameed, and Syed Saad Ali Shah. Issues in global software development (communication, coordination and trust) critical review. *training*, 6(7):8, 2016.

[70] Darja Šmite, Claes Wohlin, Tony Gorschek, and Robert Feldt. Empirical evidence in global software engineering: a systematic review. *Empirical software engineering*, 15(1):91–118, 2010.

[71] Paul L Bannerman, Emam Hossain, and Ross Jeffery. Scrum practice mitigation of global software development coordination challenges: A distinctive advantage? In *System Science (HICSS), 2012 45th Hawaii International Conference*, pages 5309–5318. IEEE, 2012.

[72] Helena Holmstrom, Eoin Ó Conchúir, Par J Agerfalk, and Brian Fitzgerald. Global software development challenges: A case study on temporal, geographical and socio-cultural distance. In *Global Software Engineering. (ICGSE), 2006 1st IEEE International Conference*, pages 3–11. IEEE, 2006.

[73] M. Jones, U. Mortensen, and Co chairmen. *Guide to software project management.* ESA Board for Software Standardisation and Control (BSSC), 1995.

[74] Mikael Lindvall and Ioana Rus. Knowledge management in software engineering. *Software, IEEE*, 19(3):26–38, 2002.

[75] Ismael Caballero, Aurora Vizcaíno, and Mario Piattini. Optimal data quality in project management for global software developments. In *Cooperation and Promotion of Information Resources in Science and Technology, 2009. COINFO'09. Fourth International Conference on*, pages 210–219. IEEE, 2009.

[76] Jinghui Zhang, Jian Chen, Junzhou Luo, and Aibo Song. Efficient location-aware data placement for data-intensive applications in geo-distributed scientific data centers. *Tsinghua Science and Technology*, 21(5):471–481, 2016.

[77] Christof Ebert and Philip De Neve. Surviving global software development. *Software, IEEE*, 18(2):62–69, 2001.

[78] Pamela Hinds. *Distributed work.* MIT Press, 2002.

[79] Mahmood Niazi, Muhammad Ali Babar, and June M Verner. Software process improvement barriers: A cross-cultural comparison. *Information and Software Technology*, 52(11):1204–1216, 2010.

[80] Elisa Hatsue Moriya Huzita, Gislaine Camila Lapasini Leal, Renato Balancieri, Tânia Fátima Calvi Tait, Edwin Cardoza, Raqueline Ritter de Moura Penteado, and Rafael Leonardo Vivian. Knowledge and contextual information management in global software development: Challenges and perspectives. In *Global Software Engineering Workshops (ICGSEW), 2012 7th IEEE International Conference*, pages 43–48. IEEE, 2012.

[81] Maarit Tihinen, Päivi Parviainen, Tanja Suomalainen, Katja Karhu, and Malin Mannevaara. Abb experiences of boosting controlling and monitoring activities in collaborative production. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference*, pages 1–5. IEEE, 2011.

[82] Quynh Mai Nguyen. *Planning in software project management: an empirical research of software companies in Vietnam.* 2006.

[83] Emam Hossain, Muhammad Ali Babar, and Hye-young Paik. Using scrum in global software development: a systematic literature review. In *Global Software Engineering (ICGSE), 2009 Fourth IEEE International Conference*, pages 175–184. IEEE Computer Society, 2009.

[84] Stefan Schneider, Richard Torkar, and Tony Gorschek. Solutions in global software engineering: A systematic literature review. *International Journal of Information Management*, 33(1):119–132, 2013.

[85] Christoph Loch and Stylianos Kavadias. *Handbook of new product development management.* Routledge, 2008.

[86] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, Mohammed Rehan Riaz, Kanaan Faisal, and Narciso Cerpa. Challenges of project management in global software development: Initial results. In *Science and Information Conference (SAI), 2013*, pages 202–206. IEEE, 2013.

[87] Aurora Vizcaíno, Félix García, José Carlos Villar, Mario Piattini, and Javier Portillo. Applying q-methodology to analyse the success factors in GSD. *Information and Software Technology*, 55(7):1200–1211, 2013.

[88] Darja Šmite, Claes Wohlin, Tony Gorschek, and Robert Feldt. Empirical evidence in global software engineering: a systematic review. *Empirical Software Engineering*, 15(1):91–118, 2010.

[89] Fabio QB da Silva, Catarina Costa, A César C França, and Rafael Prikladinicki. Challenges and solutions in distributed software development project management: A systematic literature review. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference*, pages 87–96. IEEE, 2010.

[90] Project management body of knowledge (PMBOK ® GUIDE), 3rd edition, Pennsylvania, USA. Project Management Institute, Inc, 2004.

[91] Ricardo Britto, Vitor Freitas, Emilia Mendes, and Muhammad Usman. Effort estimation in global software development: A systematic literature review. In *Proceedings of the 9th IEEE International Conference on Global Software Engineering (ICGSE)*, pages 135–144, Washington, 2014. IEEE Computer Society.

[92] Lynn Crawford. Global body of project management knowledge and standards. *The Wiley Guide to Project Organization & Project Management Competencies*, pages 206–252, 2007.

[93] *Role of standards: A guide for small and medium-sized enterprises.* United nations industrial development organization, 2006.

[94] Alain Abran, Pierre Bourque, Robert Dupuis, and James W Moore. *Guide to the software engineering body of knowledge-SWEBOK.* IEEE Press, 2001.

[95] Claude Y Laporte, Simon Alexandre, and Rory V O'Connor. A software engineering lifecycle standard for very small enterprises. pages 129–141, 2008.

[96] FIE Aust Robert Halligan. Application guidance for: ISO/IEC 15288 (IEEE std 15288 -2008). In *Systems and Software Engineering - System Life Cycle Processes*, 2008.

[97] ISO/IEC/IEEE 16326. In *Systems and Software Engineering - Life cycle processes - Project management*, 2009.

[98] Stanislaw Gasik. *Comparison of ISO 21500 and PMBOK Guide.* PhD thesis, 2009.

[99] Pearl Brereton Barbara Kitchenham. Applying Q-methodology to analyse the success factors in GSD. *Information and Software Technology*, 55:2049–2075, 2013.

[100] Darja Smite, Claes Wohlin, Robert Feldt, and Tony Gorschek. Reporting empirical research in global software engineering: a classification scheme. In *Global Software Engineering, 2008. ICGSE 2008. IEEE International Conference on*, pages 173–181. IEEE, 2008.

[101] Rosalba Giuffrida and Yvonne Dittrich. Empirical studies on the use of social software in global software development systematic mapping study. *Information and Software Technology*, 55(7):1143–164, 2013.

[102] Javier Portillo-Rodríguez, Aurora Vizcaíno, Mario Piattini, and Sarah Beecham. Tools used in global software engineering: A systematic mapping review. *Information and Software Technology*, 54(7):663–685, 2012.

[103] J Richard Landis, Gary G Koch, et al. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.

[104] Sofia Ouhbi, Ali Idri, José Luis Fernández-Alemán, and Ambrosio Toval. Requirements engineering education: a systematic mapping study. *Requirements Engineering*, pages 1–20, 2013.

[105] Adrian Fernandez, Emilio Insfran, and Silvia Abrahão. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, 53(8):789–817, 2011.

[106] Paolo Tonella, Marco Torchiano, Bart Du Bois, and Tarja Systä. Empirical studies in reverse engineering: state of the art and future trends. *Empirical Software Engineering*, 12(5):551–571, 2007.

[107] Martin Ivarsson and Tony Gorschek. A method for evaluating rigor and industrial relevance of technology evaluations. *Empirical Software Engineering*, 16(3):365–395, 2011.

[108] A guide to the project management body of knowledge. Project Management Institute, 2013.

[109] Ansgar Lamersdorf, Jurgen Munch, Alicia Fernández del Viso Torre, Carlos Rebate Sánchez, Markus Heinz, and Dieter Rombach. A rule-based model for customized risk identification in distributed software development projects. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, pages 209–218. IEEE, 2010.

[110] June M Verner, O Pearl Brereton, Barbara A Kitchenham, Mahmood Turner, and Mahmood Niazi. Risks and risk mitigation in global software development: A tertiary study. *Information and Software Technology*, 56(1):54–78, 2014.

[111] Narayan Ramasubbu and Rajesh Krishna Balan. Globally distributed software development project performance: an empirical analysis. In *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM*

SIGSOFT symposium on The foundations of software engineering, pages 125–134. ACM, 2007.

[112] Peixoto, Carlos Eduardo Lima, Jorge Luis Nicolas Audy, and Rafael Prikladnicki. Effort estimation in global software development projects: Preliminary results from a survey. In *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference*, pages 123–127. IEEE, 2010.

[113] Luis Henrique Almeida, Adriano Bessa Albuquerque, and Plácido Rogério Pinheiro. A multi-criteria model for planning and fine-tuning distributed scrum projects. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference*, pages 75–83. IEEE, 2011.

[114] Sirion Setamanit, Wayne Wakeland, and David Raffo. Planning and improving global software development process using simulation. In *Proceedings of the 2006 international workshop on Global Software Development for the practitioner*, pages 8–14. ACM, 2006.

[115] Sarah Beecham, Noel Carroll, and John Noll. A decision support system for global team management: Expert evaluation. In *Global Software Engineering Workshops (ICGSEW), 2012 7th IEEE International Conference*, pages 12–17. IEEE, 2012.

[116] Kai Stapel, Eric Knauss, Kurt Schneider, and Nico Zazworka. Flow mapping: planning and managing communication in distributed teams. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference*, pages 190–199. IEEE, 2011.

[117] Fawsy Bendeck, Sigrid Goldmann, Harald Holz, and Boris Kotting. Coordinating management activities in distributed software development projects. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, (WET ICE'98) Proceedings of the 7th IEEE International Workshops*, pages 33–38. IEEE, 1998.

[118] Paul Layzell, O Pearl Brereton, and Andrew French. Supporting collaboration in distributed software engineering teams. In *Software Engineering Conference, 2000. APSEC 2000. Proceedings of the 7th Asia-Pacific*, pages 38–45. IEEE, 2000.

[119] Danfeng Wu, Guangping Zeng, Di He, Zhaopeng Qian, and Qingchuan Zhang. Task coordination organization model and the task allocation algorithm for resource contention of the syncretic system. *Tsinghua Science and Technology*, 21 (4):459–470, 2016.

[120] Rohit M. Lotlikar, Ramana Polavarapu, Sadhika Sharma, and Biplav Srivastava. Towards effective project management across multiple projects with distributed

performing centers. In *Proceedings of the IEEE International Conference on Services Computing (SCC)*, volume 1, pages 33–40, 2008.

[121] Maria Paasivaara and Casper Lassenius. Could global software development benefit from agile methods? In *Global Software Engineering. (ICGSE), 2006 1st IEEE International Conference*, pages 109–113. IEEE, 2006.

[122] David KM Mak and PB Kruchten. Task coordination in an agile distributed software development environment. In *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference*, pages 606–611. IEEE, 2006.

[123] Zunera Jalil and Aamer Hanif. Improving management of outsourced software projects in pakistan. In *Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference*, pages 524–528. IEEE, 2009.

[124] Pedro José Garrido, Aurora Vizcaíno, Juan Andrada, Miguel J Monasor, and Mario Piattini. DPMTool: A tool for decisions management in distributed software projects. In *Global Software Engineering Workshops (ICGSEW), 2012 7th IEEE International Conference*, pages 22–27. IEEE, 2012.

[125] Ángel García-Crespo, Ricardo Colomo-Palacios, Pedro Soto-Acosta, and Marcos Ruano-Mayoral. A qualitative study of hard decision making in managing global software development teams. *Information Systems Management*, 27(3):247–252, 2010.

[126] Rafael Prikladnicki, Jorge Luis Nicolas Audy, Daniela Damian, and Toacy Cavalcante de Oliveira. Distributed software development: Practices and challenges in different business strategies of offshoring and onshoring. In *Global Software Engineering (ICGSE), 2007 2nd IEEE International Conference*, pages 262–274. IEEE, 2007.

[127] Daniela Damian. Stakeholders in global requirements engineering: Lessons learned from practice. *Software, IEEE*, 24(2):21–27, 2007.

[128] Suzanne P Weisband. *Leadership at a distance: research in technologically-supported work.* Psychology Press, 2013.

[129] Roberto Zanoni and Jorge Luis Nicolas Audy. Project management model for a physically distributed software development environment. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference*, pages 8–16. IEEE, 2003.

[130] Michael T Lane and Par J Agerfalk. On the suitability of particular software development roles to global software development. In *Global Software Engineering. (ICGSE), 2008 3rd IEEE International Conference*, pages 3–12. IEEE, 2008.

[131] Maria Paasivaara. Coaching global software development projects. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference*, pages 84–93. IEEE, 2011.

[132] Yang Xiaohu, Xu Bin, He Zhijun, and S Maddineni. Extreme programming in global software development. In *Electrical and Computer Engineering, 2004. Canadian Conference*, volume 4, pages 1845–1848. IEEE, 2004.

[133] James M Erickson and C Ranganathan. Project management capabilities: Key to application development offshore outsourcing. In *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference*, volume 8, pages 1–10. IEEE, 2006.

[134] Bjorn Forstrom Cole. An evolutionary method for synthesizing technological planning and architectural advance. 2009.

[135] James D Herbsleb, Daniel J Paulish, and Matthew Bass. Global software development at siemens: experience from nine projects. In *Software Engineering, 2005. ICSE 2005. Proceedings. 27th International Conference*, pages 524–533. IEEE, 2005.

[136] Javier García Guzmán, Javier Saldaña Ramos, Antonio Amescua Seco, and Ana Sanz Esteban. How to get mature global virtual teams: a framework to improve team process management in distributed software teams. *Software Quality Journal*, 18(4):409–435, 2010.

[137] Sarah Beecham, Padraig OLeary, Ita Richardson, Sean Baker, and John Noll. Who are we doing global software engineering research for? In *2013 IEEE 8th International Conference on Global Software Engineering*, pages 41–50. IEEE, 2013.

[138] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer, 2012. URL http://www.cs.uu.nl/docs/vakken/arm/handouts/M7-ethics.pdf.

[139] Frank Elberzhager, Jürgen Münch, and Vi Tran Ngoc Nha. A systematic mapping study on the combination of static and dynamic quality assurance techniques. *Information and Software Technology*, 54(1):1–15, 2012.

[140] Apostolos Ampatzoglou, Sofia Charalampidou, and Ioannis Stamelos. Research state of the art on GoF design patterns: A mapping study. *Journal of Systems and Software*, 86(7):1945–1964, 2013.

[141] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*, pages 285–311. Springer, 2008.

[142] Pedro Reales Mateo, Macario Polo Usaola, and Jose Luis Fernandez Aleman. Validating second-order mutation at system level. *Software Engineering, IEEE Transactions on*, 39(4):570–587, 2013.

[143] Terry Hill and Roy Westbrook. SWOT analysis: it's time for a product recall. *Long range planning*, 30(1):46–52, 1997.

[144] M. JøRgensen. A review of studies on expert estimation of software development effort. *J. Syst. Softw.*, 70(1-2):37–60, 2004.

[145] Narayan Ramasubbu and Rajesh Krishna Balan. Overcoming the challenges in cost estimation for distributed software projects. In *Proceedings of the 34th International Conference on Software Engineering (ICSE)*, pages 91–101, Piscataway, 2012. IEEE Press.

[146] Ali Idri, Fatima Azzahra Amazal, and Alain Abran. Analogy-based software development effort estimation: A systematic mapping and review. *Information and Software Technology*, 58(0):206–230, 2015.

[147] Sofia Ouhbi, Ali Idri, Jose Luis Fernández-Alemán, and Ambrosio Toval. Software quality requirements: a systematic mapping study. In *Software Engineering Conference (APSEC, 2013 20th Asia-Pacific*, volume 1, pages 231–238. IEEE, 2013.

[148] Magne Jorgensen and Martin Shepperd. A systematic review of software development cost estimation studies. *Software Engineering, IEEE Transactions on*, 33 (1):33–53, 2007.

[149] Robert T Hughes. Expert judgement as an estimating method. *Information and Software Technology*, 38(2):67–75, 1996.

[150] Ali Idri, Azeddine Zahi, and Alain Abran. Software cost estimation by fuzzy analogy for web hypermedia applications. In *Proceedings of the International Conference on Software Process and Product Measurement, Cadiz, Spain*, pages 53–62. Citeseer, 2006.

[151] Barry Boehm, Chris Abts, and Sunita Chulani. Software development cost estimation approaches survey. *Annals of Software Engineering*, 10(1-4):177–205, 2000.

[152] Ali Bou Nassif, Luiz Fernando Capretz, and Danny Ho. Software effort estimation in the early stages of the software life cycle using a cascade correlation neural network model. In *Proceedings of the 13th International Conference on Software Engineering (ACIS), Artificial Intelligence, Networking and Parallel/Distributed Computing*, pages 589–594, 2012.

[153] Mamoona Humayun and Cui Gang. Estimating effort in global software development projects using machine learning techniques. *International Journal of Information and Education Technology*, 2(3):208–211, 2012.

[154] Khaled Hamdan, Hazem El Khatib, John Moses, and Peter Smith. A software cost ontology system for assisting estimation of software project effort for use with case-based reasoning. In *Innovations in Information Technology, 2006*, pages 1–5, 2006.

[155] Nanjangud C. Narendra, Karthikeyan Ponnalagu, Nianjun Zhou, and Wesley M. Gifford. Towards a formal model for optimal task-site allocation and effort estimation in global software development. In *Proceedings of the Annual SRII Global Conference*, pages 470–477, 2012.

[156] Steve Easterbrook, Janice Singer, Margaret-Anne Storey, and Daniela Damian. Selecting empirical methods for software engineering research. In *Guide to advanced empirical software engineering*, pages 285–311. 2008.

[157] Tridas Mukhopadhyay, Steven S Vicinanza, and Michael J Prietula. Examining the feasibility of a case-based reasoning model for software effort estimation. *MIS quarterly*, pages 155–171, 1992.

[158] Lefteris Angelis, Ioannis Stamelos, and Maurizio Morisio. Building a software cost estimation model based on categorical data. In *metrics*, page 4. IEEE, 2001.

[159] Fred J Heemstra. Software cost estimation. *Information and software technology*, 34(10):627–639, 1992.

[160] Barry Boehm, Bradford Clark, Ellis Horowitz, Chris Westland, Ray Madachy, and Richard Selby. Cost models for future software life cycle processes: Cocomo 2.0. *Annals of software engineering*, 1(1):57–94, 1995.

[161] Darja Šmite and Juris Borzovs. A framework for overcoming supplier related threats in global projects. In *European Conference on Software Process Improvement*, pages 50–61. Springer, 2006.

[162] Darja Šmite. A case study: coordination practices in global software development. In *Proceedings of the 6th international conference of Product Focused Software Process Improvement (PROFES'05)*, pages 234–244. Springer, 2005.

[163] Anselm Strauss and Juliet M Corbin. *Basics of qualitative research: Grounded theory procedures and techniques.* Sage Publications, Inc, 1990.

[164] Marvin J Carr, Suresh L Konda, Ira Monarch, F Carol Ulrich, and Clay F Walker. Taxonomy-based risk identification. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 1993.

[165] Linda Hutz Pesante. Software engineering institute (sei). 2003.

[166] Gerard Oegan. Software engineering institute (sei). In *Pillars of Computing*, pages 195–205. Springer, 2015.

[167] Raymond J Madachy. Knowledge-based risk assessment and cost estimation. *Automated Software Engineering*, 2(3):219–230, 1995.

[168] Barry W Boehm, Maria H Penedo, E Don Stuckle, Robert D Williams, Arthur B Pyster, et al. A software development environment for improving productivity. In *Computer*. Citeseer, 1984.

[169] Manal El Bajta, Ali Idri, José Luis Fernández-Alemán, Joaquin Nicolas Ros, and Ambrosio Toval. Software cost estimation for global software development a systematic map and review study. In *Evaluation of Novel Approaches to Software Engineering (ENASE), 2015 International Conference on*, pages 197–206. IEEE, 2015.

[170] Sajjad Mahmood, Sajid Anwer, Mahmood Niazi, Mohammad Alshayeb, and Ita Richardson. Key factors that influence task allocation in global software development. *Information and Software Technology*, 91:102–122, 2017.

[171] Srinivas Nidhra, Muralidhar Yanamadala, Wasif Afzal, and Richard Torkar. Knowledge transfer challenges and mitigation strategies in global software development systematic literature review and industrial validation. *International journal of information management*, 33(2):333–355, 2014.

[172] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, Mohammed Rehan Riaz, Kanaan Faisal, Narciso Cerpa, Siffat Ullah Khan, and Ita Richardson. Challenges of project management in global software development: A client-vendor analysis. *Information and Software Technology*, 80:1–19, 2016.

[173] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, Abdul Majid Qureshi, Kanaan Faisal, and Narciso Cerpa. Toward successful project management in global software development. *International Journal of Project Management*, 34 (8):1553–1567, 2016.

[174] Saad Yasser Chadli, Ali Idri, Joaquín Nicolás Ros, José Luis Fernández-Alemán, Juan M Carrillo Gea, and Ambrosio Toval. Software project management tools

in global software development: a systematic mapping study. *SpringerPlus*, 5(1): 2006, 2016.

[175] Juan M Carrillo de Gea, Joaquín Nicolás, José L Fernández-Alemán, Ambrosio Toval, and Ali Idri. Are the expected benefits of requirements reuse hampered by distance? an experiment. *SpringerPlus*, 5(1):2097, 2016.

[176] Manal El Bajta, Ali Idri, Joaquín Nicolas Ros, José Luis Fernandez-Aleman, and Ambrosio Toval. Estimation of costs and time for the development of distributed software. In *Software Project Management for Distributed Computing*, pages 25–42. Springer, 2017.

[177] Parastoo Mohagheghi. Global software development: Issues, solutions, challenges. *Retrieved April*, 7:2009, 2004.

[178] Ricardo Britto, Emilia Mendes, and Jürgen Börstler. An empirical investigation on effort estimation in agile global software development. In *2015 IEEE 10th International Conference on Global Software Engineering*, pages 38–45. IEEE, 2015.

[179] Ansgar Lamersdorf and Jürgen Münch. A multi-criteria distribution model for global software development projects. *Journal of the Brazilian Computer Society*, 16(2):97–115, 2010.

[180] Barbara Kitchenham. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26, 2004.

[181] Raymond J Madachy. Cost modeling of distributed team processes for global development and software-intensive systems of systems. *Software Process: Improvement and Practice*, 13(1):51–61, 2008.

[182] Robert M Groves, Floyd J Fowler Jr, Mick P Couper, James M Lepkowski, Eleanor Singer, and Roger Tourangeau. *Survey methodology*, volume 561. John Wiley & Sons, 2011.

[183] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, 14(2):131, 2009.

[184] Nancy Thayer-Hart, Jennifer Dykema, Kelly Elver, Nora Cate Schaeffer, and John Stevenson. Survey fundamentals: A guide to designing and implementing surveys. *Office of Quality Improvement*, 2010.

[185] Manal El Bajta. Analogy-based software development effort estimation in global software development. In *IEEE 10th International Conference on Global Software Engineering Workshops (ICGSEW)*, pages 51–54. IEEE, 2015.

[186] Carroll M Graham and Fredrick Muyia Nafukho. Culture, organizational learning and selected employee background variables in small-size business enterprises. *Journal of European Industrial Training*, 31(2):127–144, 2007.

[187] Barbara A Kitchenham, Shari Lawrence Pfleeger, Lesley M Pickard, Peter W Jones, David C. Hoaglin, Khaled El Emam, and Jarrett Rosenberg. Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on software engineering*, 28(8):721–734, 2002.

[188] Barbara Kitchenham and Shari Lawrence Pfleeger. Principles of survey research part 6: data analysis. *ACM SIGSOFT Software Engineering Notes*, 28(2):24–27, 2003.

[189] Tore Dybå, Vigdis By Kampenes, and Dag IK Sjøberg. A systematic review of statistical power in software engineering experiments. *Information and Software Technology*, 48(8):745–755, 2006.

[190] J Alberto Espinosa and Erran Carmel. The effect of time separation on coordination costs in global software teams: A dyad model. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2004.

[191] Darja Smite, Fabio Calefato, and Claes Wohlin. Cost savings in global software engineering. *IEEE Software*, (4):26–32, 2015.

[192] Zhihao Chen, Tim Menzies, Dan Port, and Barry Boehm. Feature subset selection can improve software cost estimation accuracy. In *ACM SIGSOFT Software Engineering Notes*, volume 30, pages 1–6. ACM, 2005.

[193] Uwe van Heesch and Paris Avgeriou. Mature architecting-a survey about the reasoning process of professional architects. In *9th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, pages 260–269. IEEE, 2011.

[194] Donald Firesmith. Common requirements problems, their negative consequences, and the industry best practices to help solve them. *Journal of Object Technology*, 6(1):17–33, 2007.

[195] Robert L Glass and Iris Vessey. Contemporary application-domain taxonomies. *IEEE Software*, (4):63–76, 1995.

[196] Dorina C Gumm. Distribution dimensions in software development projects: A taxonomy. *IEEE software*, 23(5):45–51, 2006.

[197] Dorina C Gumm. The phenomenon of distribution in software development projects: A taxonomy proposal. In *Proc. European Mediterranean Conf. Information Systems*. Citeseer, 2005.

[198] Paula Laurent, Patrick Mader, Jane Cleland-Huang, and Adam Steele. A taxonomy and visual notation for modeling globally distributed requirements engineering projects. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 35–44. IEEE, 2010.

[199] Darja Šmite, Claes Wohlin, Zane Galviņa, and Rafael Prikladnicki. An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, 19(1):105–153, 2014.

[200] Muhammad Usman, Ricardo Britto, Jürgen Börstler, and Emilia Mendes. Taxonomies in software engineering: A systematic mapping study and a revised taxonomy development method. *Information and Software Technology*, 85:43–59, 2017.

[201] Barbara H Kwasnik. The role of classification in knowledge representation and discovery. 1999.

[202] Tore Dybå, Dag IK Sjøberg, and Daniela S Cruzes. What works for whom, where, when, and why?: on the role of context in empirical software engineering. In *Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement*, pages 19–28. ACM, 2012.

[203] Anind K Dey. Understanding and using context. *Personal and ubiquitous computing*, 5(1):4–7, 2001.

[204] Hannu Jaakkola. Towards a globalized software industry. *Acta Polytechnica Hungarica*, 6(5):69–84, 2009.

[205] Narayan Ramasubbu, Marcelo Cataldo, Rajesh Krishna Balan, and James D Herbsleb. Configuring global software teams: a multi-company analysis of project productivity, quality, and profits. In *Software Engineering (ICSE), 2011 33rd International Conference on*, pages 261–270. IEEE, 2011.

[206] Par J Agerfalk, Brian Fitzgerald, Helena Holmstrom Olsson, Brian Lings, Bjorn Lundell, and Eoin Ó Conchúir. A framework for considering opportunities and threats in distributed software development. 2005.

[207] Lawrence Chung, Brian A Nixon, Eric Yu, and John Mylopoulos. *Non-functional requirements in software engineering*, volume 5. Springer Science & Business Media, 2012.

[208] James Cusick and Alpana Prasad. A practical management and engineering approach to offshore collaboration. *Software, IEEE*, 23(5):20–29, 2006.

[209] Amrit Tiwana. Beyond the black box: knowledge overlaps in software outsourcing. *Software, IEEE*, 21(5):51–58, 2004.

[210] Kevin C Desouza, Yukika Awazu, and Peter Baloh. Managing knowledge in global software development efforts: Issues and practices. *Software, IEEE*, 23(5):30–37, 2006.

[211] Richard Heeks, Srinivas Krishna, B Nicholsen, and Sundeep Sahay. Synching or sinking: global software outsourcing relationships. *Software, IEEE*, 18(2):54–60, 2001.

[212] Narayan Ramasubbu, Mayuram S Krishnan, and Prasad Kompalli. Leveraging global resources: A process maturity framework for managing distributed development. *Software, IEEE*, 22(3):80–86, 2005.

[213] Anh Nguyen-Duc, Daniela S Cruzes, and Reidar Conradi. The impact of global dispersion on coordination, team performance and software quality–a systematic literature review. *Information and Software Technology*, 57:277–294, 2015.

[214] Sameer Patil, Alfred Kobsa, Ajita John, and Doree Seligmann. Methodological reflections on a field study of a globally distributed software project. *Information and Software Technology*, 53(9):969–980, 2011.

[215] Seiyoung Lee and Hwan-Seung Yong. Distributed agile: project management in a global environment. *Empirical Software Engineering*, 15(2):204–217, 2010.

[216] Rafael Prikladnicki, Alexander Boden, Gabriela Avram, Cleidson RB de Souza, and Volker Wulf. Data collection in global software engineering research: learning from past experience. *Empirical Software Engineering*, 19(4):822–856, 2014.

[217] Helena Holmström, Brian Fitzgerald, Pär J Ågerfalk, and Eoin Ó Conchúir. Agile practices reduce distance in global software development. *Information Systems Management*, 23(3):7–18, 2006.

[218] Adel Taweel, Brendan Delaney, Theodoros N Arvanitis, and Lei Zhao. Communication, knowledge and co-ordination management in globally distributed software development: Informed by a scientific software engineering case study. In *Global Software Engineering, 2009. ICGSE 2009. 4th IEEE International Conference*, pages 370–375. IEEE, 2009.

[219] Ansgar Lamersdorf, Jürgen Munch, and Dieter Rombach. A survey on the state of the practice in distributed software development: Criteria for task allocation. In *Global Software Engineering, 2009. (ICGSE) 2009. 4th IEEE International Conference*, pages 41–50. IEEE, 2009.

[220] Gabriela Avram. Of deadlocks and peopleware-collaborative work practices in global software development. In *Global Software Engineering (ICGSE), 2007 2nd IEEE International Conference*, pages 91–102. IEEE, 2007.

[221] Christian Bartelt, Manfred Broy, Christoph Herrmann, Eric Knauss, Marco Kuhrmann, Andreas Rausch, Bernhard Rumpe, and Kurt Schneider. Orchestration of global software engineering projects-position paper. In *Global Software Engineering (ICGSE), 2009 4th IEEE International Conference*, pages 332–337. IEEE, 2009.

[222] Suling Zhang, Marilyn Tremaine, Allen E Milewski, and Felix Kobler. Delegation in global software teams: Leading or managing? In *Global Software Engineering, 2009. ICGSE 2009 4th IEEE International Conference*, pages 293–294. IEEE, 2009.

[223] Christian R Prause, Marius Scholten, Andreas Zimmermann, René Reiners, and Markus Eisenhauer. Managing the iterative requirements process in a multinational project using an issue tracker. In *Global Software Engineering. (ICGSE), 2008 3rd IEEE International Conference*, pages 151–159. IEEE, 2008.

[224] Martin Nordio, H-C Estler, Bertrand Meyer, Julian Tschannen, Carlo Ghezzi, and Elisabetta Di Nitto. How do distribution and time zones affect software development? a case study on communication. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference*, pages 176–184. IEEE, 2011.

[225] M. Fernandez and J. Basavaraju. Task allocation model in globally distributed software projects using genetic algorithms. In *Global Software Engineering (ICGSE), 2012 7th IEEE International Conference*, page 181. IEEE, 2012.

[226] Kate Ehrlich and Klarissa Chang. Leveraging expertise in global software teams: Going outside boundaries. *Global Software Engineering (ICGSE), 2006 1st IEEE International Conference*, 6:149–158, 2006.

[227] Sajjad Mahmood, Mahmood Niazi, and Akhtar Hussain. Identifying the challenges for managing component-based development in global software development: Preliminary results. In *Science and Information Conference (SAI), 2015*, pages 933–938. IEEE, 2015.

[228] Nelson Denhere, Tertia Hörne, and John Andrew van der Poll. Managing globally distributed software development projects using virtual teams: A middle east case study. In *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists*, page 12. ACM, 2015.

[229] Frank Salger, Gregor Engels, and Alexander Hofmann. Assessments in global software development: a tailorable framework for industrial projects. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pages 29–38. ACM, 2010.

[230] Siva Dorairaj, James Noble, and Petra Malik. Knowledge management in distributed agile software development. In *Agile Conference (AGILE), 2012*, pages 64–73. IEEE, 2012.

[231] Julia Kotlarsky, Paul C Van Fenema, and Leslie P Willcocks. Developing a knowledge-based perspective on coordination: The case of global software projects. *Information & Management*, 45(2):96–108, 2008.

[232] Debasisha Mishra and Biswajit Mahanty. The effect of onsite-offshore work division on project cost, schedule, and quality for re-engineering projects in indian outsourcing software industry. *Strategic Outsourcing: An International Journal*, 7 (3):198–225, 2014.

[233] Alexander Boden and Gabriela Avram. Bridging knowledge distribution-the role of knowledge brokers in distributed software development teams. In *Cooperative and Human Aspects on Software Engineering, 2009. CHASE'09. ICSE Workshop*, pages 8–11. IEEE, 2009.

[234] Anum Tariq and Aliya Ashraf Khan. Framework supporting team and project activities in global software development (GSD). In *Emerging Technologies (ICET), 2012 International Conference*, pages 1–6. IEEE, 2012.

[235] Maria Paasivaara, Casper Lassenius, Daniela Damian, Petteri Räty, and Adrian Schröter. Teaching students global software engineering skills using distributed scrum. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1128–1137. IEEE, 2013.

[236] Hai Eric Lam and Piyush Maheshwari. Task and team management in the distributed software project management tool. In *Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International*, pages 401–408. IEEE, 2001.

[237] Maya Daneva, Egbert Van Der Veen, Chintan Amrit, Smita Ghaisas, Klaas Sikkel, Ramesh Kumar, Nirav Ajmeri, Uday Ramteerthkar, and Roel Wieringa. Agile requirements prioritization in large-scale outsourced system projects: An empirical study. *Journal of Systems and Software*, 86(5):1333–1353, 2013.

[238] Matteo Gaeta and Pierluigi Ritrovato. Generalised environment for process management in cooperative software engineering. In *Computer Software and Applications Conference, COMPSAC 2002. Proceedings. 26th Annual International*, pages 1049–1053. IEEE, 2002.

[239] Bin Xu. A service oriented model for role based global cooperative software development. In *Convergence Information Technology, 2007, International Conference*, pages 376 – 381. IEEE, 2007.

[240] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, Abdul Majid Qureshi, Kanaan Faisal, and Narciso Cerpa. Towards identifying the factors for project management success in global software development: Initial results. In *ICSEA 2013, The 8th International Conference on Software Engineering Advances*, pages 285–290, 2013.

[241] Alan Moraes, Eduardo Silva, Cleyton da Trindade, Yuri Barbosa, and Silvio Meira. Recommending experts using communication history. In *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*, pages 41–45. ACM, 2010.

[242] J. Makio and S. Betz. On educating globally distributed software development case study. In *Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium*, pages 480–485. IEEE, 2009.

[243] Daniela Damian, Casper Lassenius, Maria Paasivaara, Arber Borici, and A Schroter. Teaching a globally distributed project course using Scrum practices. In *Collaborative Teaching of Globally Distributed Software Development Workshop (CTGDSD), 2012*, pages 30–34. IEEE, 2012.

[244] Ian Broooks and Kathleen Swigger. The role of leadership and its effect on the temporal patterns of global software development teams. In *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2012 8th International Conference*, pages 381–390. IEEE, 2012.

[245] Mahmood Niazi, Sajjad Mahmood, Mohammad Alshayeb, and Ayman Hroub. Empirical investigation of the challenges of the existing tools used in global software development projects. *Software, IET*, 9(5):135–143, 2015.

[246] Torgeir Dingsoyr and Darja Smite. Managing knowledge in global software development projects. *IT Professional*, 16(1):22–29, 2014.

[247] Lucas D Panjer, Daniela Damian, and Margaret-Anne Storey. Cooperation and coordination concerns in a distributed software development project. In *Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pages 77–80. ACM, 2008.

[248] Luis Henrique Almeida, Plácido Rogério Pinheiro, and Adriano Bessa Albuquerque. Applying multi-criteria decision analysis to global software development with scrum project planning. In *Rough Sets and Knowledge Technology*, pages 311–320. Springer, 2011.

[249] Tamara Sulaiman, Brent Barton, and Thomas Blackburn. Agile EVM-earned value management in scrum projects. In *Agile Conference, 2006*, pages 10–16. IEEE, 2006.

Appendix A

Detailed results of the SMS presented in Chapter 2

Appendix A provides detailed results of the SMS presented in Chapter 2.

Table A.1 shows the results per source after the inclusion and exclusion process.

Table A.2 shows a five-level quality classification.

Table A.3 shows the publication channels.

Table A.4 presents the journals and conferences in which the papers selected for this SMS were published.

Table A.6 presents the list of the selected papers with detailed information on their classification results and quality assessment.

Table A.7 provides a summary of SPM activities for GSD.

Table A.8 emphasises the benefits and limits of the main approaches in SPM for GSD.

Table A.1: Source results

Sources	Relevant studies	Selected studies
IEEE Digital Library	133	52
ACM Digital Library	52	15
Science Direct	28	9
SpringerLink	5	3
Google scholar	21	5
Total	239	84

Table A.2: Quality levels of relevant studies

Quality level	Papers	Percent (%)
Very high ($4 < score \leqslant 5$)	13	15.5
High ($3 < score \leqslant 4$)	26	30.9
Medium ($2 < score \leqslant 3$)	24	28.6
Low ($1 < score \leqslant 2$)	19	22.6
Very low ($0 < score \leqslant 1$)	2	2.4

Table A.3: Publication channel

Publication channels	Selected papers	Percent (%)
Conference	50	59.5
Journal	29	34.5
Workshop	5	6
Total	84	100

TABLE A.4: Distribution of selected studies: Journal (J) and Conferences (C)

Publication channels	Type	Papers	Total
IEEE Software	J	[208], [209], [210], [77], [211], [212]	6
Information and Software Technology	J	[8], [213], [101], [87], [214], [79]	5
Empirical Software Engineering	J	[88], [215], [216]	3
Information Systems Management	J	[125] , [217]	2
International Conference on Global Software Engineering (ICGSE)	C	[113], [218], [219], [220], [89], [221], [112], [130], [222], [116], [223], [131], [224], [225], [81], [72], [121], [226], [218], [178]	20
Electrical and Computer Engineering (CCECE)	C	[132], [122]	2
Annual Hawaii International Conference on System Science	C	[129], [133]	2
International Conference on Software Engineering (APSEC)	C	[135], [118]	2
Science and Information Conference (SAI)	C	[227], [228]	2
Others	—	—	15

TABLE A.5: Acronyms used in Table A.6

Acronyms					
Research Type		Research Method		Approach	
Evaluation Research	E.R.	Case Study	C.S.	Data Mining Technique	D.M.T.
Solution Proposal	S.P.	Experiment	Exp.	Framework	F.
Experience	Expe.	Survey	Sur.	-	-

Table A.6: Mapping questions and quality assessment

Paper	Classification					Quality assessment				
	Ch.	Year	Research Type	Research method	Approach	(a)	(b)	(c)	(d)	Score
[101]	J.	2013	Review	C.S.	Method	2	1	1	1	5
[210]	J.	2006	E.R.	Review	Model	2	1	1	1	5
[209]	J.	2004	S.P.	C.S.	Process	2	1	1	1	5
[212]	J.	2005	S.P.	C.S.	Process	2	1	1	1	5
[77]	J.	2001	Expe.	C.S.	Method	2	1	1	1	5
[24]	J.	2003	E.R.	Sur.	Method	2	1	1	1	5
[87]	J.	2013	E.R.	Sur.	F.	2	1	1	1	5
[214]	J.	2011	E.R.	C.S.	Method	2	1	1	1	5
[84]	J.	2013	S.P.	C.S.	Model	1.5	1	1	1	4.5
[135]	C.	2005	Expe.	C.S.	Method	1.5	1	1	1	4.5
[171]	J.	2014	S.P.	C.S.	Method	1.5	1	1	1	4.5
[199]	J.	2014	Expe.	C.S.	Method	1.5	1	1	1	4.5
[215]	J.	2010	S.P.	C.S.	Method	1.5	1	1	1	4.5
[8]	J.	2006	Expe.	C.S.	Method	2	1	0	1	4
[208]	J.	2006	Expe.	C.S.	F.	2	1	0	1	4
[111]	C.	2007	S.P.	C.S.	D.M.T.	1	1	1	1	4
[229]	C.	2010	E.R.	C.S.	Method	1	1	1	1	4
[230]	C.	2012	E.R.	C.S.	Method	1	1	1	1	4
[213]	J.	2012	E.R.	Review	F.	2	1	1	0	4
[231]	J.	2008	E.R.	C.S.	Method	1	1	1	1	4
[216]	J.	2014	S.P.	C.S.	Model	1	1	1	1	4
[232]	J.	2014	Expe.	C.S.	Method	2	1	0	1	4
[232]	J.	2014	Expe.	C.S.	F.	2	1	0	1	4
[226]	C.	2006	Expe.	C.S.	D.M.T.	0.5	1	1	1	3.5
[124]	W.	2012	E.R.	C.S.	D.M.T.	0.5	1	1	1	3.5
[113]	C.	2011	S.P.	C.S.	Method	0.5	1	1	1	3.5
[72]	C.	2006	E.R.	C.S.	Process	0.5	1	1	1	3.5
[121]	C.	2006	E.R.	C.S.	Method	0.5	1	1	1	3.5
[223]	C.	2008	E.R.	C.S.	Process	0.5	1	1	1	3.5
[131]	C.	2011	Expe.	Exp.	Method	0.5	1	1	1	3.5
[133]	C.	2006	E.R.	C.S.	F.	1.5	1	0	1	3.5

Paper	Classification					Quality assessment				
	Ch.	Year	Research Type	Research method	Approach	(a)	(b)	(c)	(d)	Score
[112]	C.	2010	E.R.	Sur.	D.M.T.	0.5	1	1	1	3.5
[116]	C.	2011	E.R.	C.S.	Method	0.5	1	1	1	3.5
[220]	C.	2007	E.R.	C.S.	Other	0.5	1	1	1	3.5
[122]	C.	2006	S.P.	C.S.	D.M.T.	0.5	1	1	1	3.5
[233]	W.	2009	E.R.	Sur.	Method	1.5	1	0	1	3.5
[89]	C.	2010	E.R.	Sur.	Model	0.5	1	1	1	3.5
[234]	C.	2012	S.P.	C.S.	F.	0.5	1	1	1	3.5
[114]	C.	2006	E.R.	Sur.	Model	0	1	1	1	3
[136]	J.	2010	E.R.	Exp.	Method	1	1	0	1	3
[235]	C.	2006	Expe.	C.S.	Method	1	1	0	1	3
[211]	J.	2001	E.R.	C.S.	F.	2	0	0	1	3
[29]	C.	2013	S.P.	C.S.	D.M.T.	0	1	1	1	3
[236]	C.	2001	E.R.	C.S.	Model	1	1	1	0	3
[228]	C.	2015	E.R.	C.S.	F.	0	1	1	1	3
[227]	C.	2015	Review	Theory	Other	0	1	1	1	3
[79]	J.	2010	E.R.	Sur.	Process	2	0	0	1	3
[115]	W.	2012	S.P.	Exp.	Model	0	1	1	1	3
[237]	J.	2013	E.R.	C.S.	Method	1.5	0	0	1	2.5
[178]	C.	2015	E.R.	Exp.	Method	0.5	1	0	1	2.5
[218]	C.	2009	E.R.	C.S.	Model	0.5	1	1	0	2.5
[219]	C.	2009	E.R.	Sur.	Method	0.5	1	0	1	2.5
[238]	C.	2002	E.R.	C.S.	Other	1	0.5	0	1	2.5
[88]	J.	2010	E.R.	Theory	Process	1.5	0	0	1	2.5
[239]	C.	2007	S.P.	C.S.	Method	0.5	1	0	1	2.5
[21]	J.	2009	E.R.	Theory	Model	1.5	0	0	1	2.5
[25]	C.	2006	E.R.	C.S.	D.M.T.	0.5	1	0	1	2.5
[240]	C.	2013	E.R.	C.S.	Model	0.5	1	0	1	2.5
[132]	C.	2004	E.R.	C.S.	Method	0.5	1	0	1	2.5
[224]	C.	2011	Expe.	C.S.	Process	0.5	1	0	1	2.5
[81]	C.	2011	E.R.	C.S.	Model	0.5	1	0	1	2.5
[118]	C.	2000	E.R.	Theory	Method	0.5	1	1	0	2.5
[241]	C.	2010	E.R.	C.S.	D.M.T.	0	1	0	1	2

Paper	Classification					Quality assessment				
	Ch.	Year	Research Type	Research method	Approach	(a)	(b)	(c)	(d)	Score
[113]	C.	2011	E.R.	C.S.	Model	0	1	0	1	2
[162]	C.	2005	E.R.	C.S.	Method	1	0	0	1	2
[221]	C.	2009	E.R.	Sur.	Process	0.5	0.5	0	1	2
[242]	C.	2009	E.R.	C.S.	Method	0.5	0.5	0	1	2
[120]	C.	2008	E.R.	C.S.	Other	0.5	0.5	0	1	2
[243]	W.	2012	E.R.	Exp.	Method	0	1	0	1	2
[244]	C.	2012	E.R.	C.S.	Other	0	1	0	1	2
[123]	C.	2009	E.R.	C.S.	Method	0	1	0	1	2
[129]	C.	2003	S.P.	C.S.	Model	0	1	0	1	2
[117]	C.	1998	E.R.	C.S.	Model	0	1	0	1	2
[245]	J.	2015	E.R.	C.S.	Method	1	1	0	0	2
[125]	J.	2010	E.R.	C.S.	D.M.T.	0.5	0	0	1	1.5
[130]	C.	2008	E.R.	C.S.	Other	0.5	0	0	1	1.5
[246]	J.	2013	E.R.	Exp.	Method	0.5	1	0	0	1.5
[218]	C.	2009	Other	Theory	Other	0.5	1	0	0	1.5
[222]	C.	2009	E.R.	C.S.	Method	0.5	0	0	1	1.5
[225]	C.	2012	S.P.	C.S.	D.M.T.	0.5	1	0	0	1.5
[80]	W.	2007	E.R.	Exp.	Process	0.5	0	1	0	1.5
[247]	C.	2008	E.R.	Sur.	Method	0	1	0	0	1
[75]	C.	2009	E.R.	C.S.	Model	0	1	0	0	1

Table A.7: SPM activities for GSD

Knowledge area	Activities	No of papers	Papers
Project integration management	Coordination Organization Decision management	13	[124], [239], [221], [231], [122], [234], [81], [162], [247], [21], [231], [225], [228]
Project scope management	Knowledge management Requirement management	8	[80], [209], [210], [230], [210], [237], [242], [233], [246]
Project time management	Planning and scheduling Time estimation	7	[208], [113], [114], [120], [232], [117], [248],
Project cost management	Cost estimation Effort estimation	8	[112], [25], [226], [248], [178], [214], [211], [29]
Project human resource management	Collaboration Team management	10	[219], [130], [222], [133], [136], [77], [115], [131], [244], [125]
Project communication management	Communication Cooperation	19	[218], [220], [238], [223], [224], [215], [135], [241], [24], [243], [118], [116], [72], [8], [121], [132], [101], [236], [228]

Table A.8: Benefits and limits of the use of approaches in SPM for GSD (MQ7)

Type	Approach	Benefits (+) and limits (-)	Papers
Method	Agile based management method: *Scrum* *Extreme programming* *Feature Driven Development* *Crystal*	(+) Facilitates task coordination and allocation decisions. (+) Brings transparency of work progress to all partners and provides a good picture of how the project is progressing. (-) Difficulties in transferring the context to different projects even within the same organisation.	[113], [243], [230], [122], [246], [215], [8], [121], [132], [237], [217]
	Flow mapping	(+) Incorporates communication at all stages of the project. (+) Makes it easy to plan and manage communication, especially when team building takes place. (-) Becomes out of date (as regards maintaining the visualisation of the Flow map) and the execution of conformance analysis is very expensive.	[116]
	Change management method	(+) Helps to organise and track development work. (+) Provides mechanisms with which to version the code and some ability to manage concurrent changes in a structured way. (+) Makes it easy to find and contact an appropriate expert, using change history data. (-) Slows the work down.	[24]
	VTManager	(+) Provides a set of effective guidelines and activities for training, developing and managing. (+) Can be adapted to different circumstances that usually arise in global software projects. (+) Determines effectiveness and efficiency: gathers and analyses data in the time needed for preparation and launch (-) Difficulties in applying it to different types of teams.	[136]
	Knowledge Management	(+) Simplifies the process of sharing, distributing, creating, capturing and understanding the company's knowledge (+) Allows team members to revisit and further understand the data at a later time. (+) Provides correct and complete understanding of the needs, and effectively contributes to the growth and utilisation of knowledge. (-) The uninterrupted commitment of the team is vital for cooperation.	[233], [230]
	Earned value management (EVM)	(+)Measures project performance and progress. (+) Presents important participants to administration of project contracts. (-) Requires sufficient project management knowledge, training and experience.	[123], [249]
Model	Estimation models: *Planning Poker based model* *Function points based model* *Use case points based model*	(+) Permits the recording of difficult decisions that must be made in the management of GSD. (-) It is not possible to correlate the estimation model in use with the number of locations involved in a distributed project	[112], [29]
	Multi-criteria decision model	(+) Describes the interactive process of modelling used to develop the project in detail. (+) Helps to facilitate forward movement during group discussions. (-) There is a lack of information on the globality of the tasks when dividing the project into independent tasks.	[113], [248]
	CMMI	(+) Analyses potential Causes of Weaknesses and defines Improvement Goals. (-) Needs good awareness of the problems to find solutions and design a new model.	[75], [123], [81]
	COCOMO	(+) Minimises cross-site communication and facilitates communication among remote collaborating teams. (+) Widely used and accepted internationally and by organisations of all sizes for estimating cost of software projects. (-) Needs an extra effort for the understanding of what is behind the data.	[111], [25]

Type	Approach	Benefits (+) and limits (-)	Papers
	Procura	(+) Allows planning and scheduling of agent-based design projects in a hierarchical top-down approach.	[117]
	CoMo-Kit	(+) Improves the processes by means of descriptive modeling and implements methodologies for project planning.	[117]
	Global Teaming Model	(+) Ensures consistency and compatibility among recommendations, and avoids conflicting strategies. (-) The knowledge base of the model needs more refinement.	[115]
	GSD model	(+) Provides the ability to explicitly represent the process structure and mechanisms used to transfer work products and to coordinate activities in GSD.	[242]
	Hybrid simulation model	(+) Exploits the efficiencies of standard schemas and representations to support project planning and process improvement in distributed projects. (-) Needs real-world data in order to calibrate the model to a specific project.	[210], [114]
Process	Delphi	(+) Collects responses and factors from the difficult decisions identified in GSD management (assigning work packages, choosing coordination mechanisms and tools, selecting internal personnel). (+) Attempts to obtain a consensus from a group of experts using repeated responses from questionnaires and controlled feedback. (-) A meaningful group of experts must be identified and managed in time.	[125]
	Software Process Improvement	(+) Helps organisations to develop higher-quality software.	[230]
Data mining technique	quadratic assignment procedure (QAP)	(+) Runs the correlations and multiple regression analyses as regards communication. (-) Provides sensitive results for particular methods and options implemented in standard software packages.	[226]
	GA and MOEA	(+) Evolve the most optimum allocation pattern considering the project goals. (+) Help the project manager by balancing various objectives and generating sets of optimised schedules for each individual team member. (-) Need to be enhanced for multiple project situations	[122], [225]
	Fuzzy similarity	(+) Determines the similarity score among the knowledge components. (-) Similarity measures are affected by irrelevant factors, thus decreasing the estimation accuracy.	[241]
	Case-Based Reasoning (CBR)	(+) Solves problems of application of knowledge in GSD using previous solutions stored in the system (-) Lack of flexibility of the knowledge representation.	[124]
Framework	Resource Based View (RBV)	(+) Identifies the key project management capabilities associated with offshore application development. (+) Provides richness and depth of information. (-) Has a limited ability to provide any reliable predictions.	[133]
	Dynamic Capabilities	(+) Improves the resources to better meet the needs of a changing competitive environment.	[133]
	Information Quality Management Framework (IQMF)	(+) Assessment and improvement of the data/information quality within the GSD Project Management	[75]