

RESEARCH

Open Access



An extended global software engineering taxonomy

Ricardo Britto*, Claes Wohlin and Emilia Mendes

*Correspondence:
ricardo.britto@bth.se
Department of Software
Engineering, Blekinge Institute of
Technology, 371 79 Karlskrona,
Sweden

Abstract

Background: In Global Software Engineering (GSE), the need for a common terminology and knowledge classification has been identified to facilitate the sharing and combination of knowledge by GSE researchers and practitioners. A GSE taxonomy was recently proposed to address such a need, focusing on a core set of dimensions; however its dimensions do not represent an exhaustive list of relevant GSE factors. Therefore, this study extends the existing taxonomy, incorporating new GSE dimensions that were identified by means of two empirical studies conducted recently.

Methods: To address the research questions of this study, we used evidence found through a systematic literature review and a survey. Based on literature, new dimensions were added to the existing taxonomy.

Results: We identified seven dimensions to extend and incorporate into the recently proposed GSE taxonomy. The resulting extended taxonomy was later on validated by comparing it with the existing taxonomy on which the extension is built and one additional taxonomy. We also demonstrated the utility of the extended taxonomy using it to classify eight finished real GSE projects. The extended taxonomy was representative enough to classify the projects in a clear way.

Conclusions: The extended taxonomy can help both researchers and practitioners by providing dimensions that can enable the description of different GSE contexts in a more comprehensive way; this can facilitate the understanding, comparison and aggregation of GSE-related findings.

Keywords: Software engineering management, Global software engineering, Knowledge classification, Taxonomy

1 Introduction

1.1 Context

Throughout the years, the software industry has applied many different software development approaches, aiming at increasing process efficiency and profitability. Numerous companies worldwide develop software in a globally distributed manner (Global Software Engineering - GSE) to achieve benefits such as reduced time-to-market and access to skillful people all over the world (Ramasubbu et al. 2011; Conchúir et al. 2009; Bondi and Ros 2009; Herbsleb and Moitra 2001).

Despite all the benefits argued to be achieved by means of GSE, it comes with challenges. These challenges often impact the productivity (Herbsleb et al. 2000) and effectiveness (Espinosa et al. 2007) of distributed software development, leading to delayed projects (Herbsleb and Mockus 2003; Šmite 2006).

The considerable number of delayed projects reported in literature indicates that practitioners have fallen short of providing accurate and reliable effort estimates in both collocated and globally distributed projects. To better understand these challenges, two studies related to effort estimation in GSE were carried out (Britto et al. 2014; Britto et al. 2015). These two studies among other findings confirmed the results reported by others, e.g. (Šmite et al. 2014), i.e. that there is a lack of a common terminology in GSE, which makes it hard to compare and synthesize results across studies.

1.2 Problem outline

Britto et al. (2014) conducted a systematic literature review (SLR) on effort estimation in the GSE context aiming at identifying the particularities of effort estimation in GSE projects. However, despite the relevance of this research topic, Britto et al. identified just a few studies supported by empirical evidence. In addition, the authors also found out that the related studies are reported in an ad-hoc manner, i.e. no common terminology or knowledge organization scheme was used to report effort-related studies in the context of GSE projects. The absence of a common terminology and structured knowledge organization can hinder the understanding of studies' contexts, making the studies harder to analyze and compare as well as aggregating the results from similar studies. Thus, it can hinder the advances in the field and the transfer of research results to industry. A classification scheme can mitigate the aforementioned problems (Vegas et al. 2009).

In the context of GSE, it has been common to use taxonomies as classification schemes to organize the existing knowledge in the field (Gumm 2006; Laurent et al. 2010; Šmite et al. 2014). According to the Oxford English Dictionary (Dictionaries 2010), a taxonomy is "a scheme of classification". This concept was initially devised to classify organisms (Linnaeus 1758), although it has been applied in many different domains, e.g. education (Bloom 1956), psychology (Moffitt 1993) and computer science (Scharstein and Szeliski 2002).

Originally, the taxonomy approach was designed to classify knowledge in a hierarchical way. Nevertheless, to date many different classification structures have been used to construct taxonomies, e.g. "hierarchy", "tree" and "facet-based" (Kwasnik 1999).

A classification scheme, such as a taxonomy, can be beneficial for both researchers and practitioners in four different ways:

1. It can ease the sharing of knowledge (Vessey et al. 2005; Vegas et al. 2009; Wohlin 2014).
2. It can help to identify gaps in a particular knowledge area (Vessey et al. 2005; Vegas et al. 2009; Wohlin 2014).
3. It can provide a better understanding of the interrelationships between the factors associated to a particular knowledge area (Vegas et al. 2009).
4. It can support decision making processes (Vegas et al. 2009).

1.3 Objective

Recently, Smite et al. proposed a GSE taxonomy, but their proposal does not include an exhaustive list of GSE factors; a taxonomy is a classification scheme that is expected to evolve over time (Vegas et al. 2009). Therefore, the main goal of this paper is to extend Smite et al.'s taxonomy, adding new factors that were identified by means of two empirical studies, i.e. an SLR (Britto et al. 2014) and a survey (Britto et al. 2015).

1.4 Contribution

We achieved the objectives of this paper with the following contributions:

- An extended GSE taxonomy based on evidence from an SLR and a survey.
- A validation of the proposed extended GSE taxonomy comparing it with other similar taxonomies and demonstrating its utility through classifying eight finished real GSE projects.

1.5 Outline

The remainder of this paper is organized as follows. A discussion of the related work is presented in Section 2. Section 3 explains the applied research design and methodology. Section 4 presents the extended GSE taxonomy, followed by its validation in Section 5. Section 6 provides a discussion of the academic/industrial implications, which is followed by the discussion of the limitations of this work in Section 7. Finally, in Section 8 we draw our conclusions and present directions for future work.

2 Background and literature review

We identified three taxonomies (Gumm 2006; Laurent et al. 2010; Šmite et al. 2014) and two ontologies (Vizcaíno et al. 2012; Marques et al. 2013) that provide knowledge organization schemes in the GSE context. It is important to note that other taxonomies were also identified (Ågerfalk and Fitzgerald 2008; Robinson and Kalakota 2010; Hofner et al. 2011), but they were already used as basis for Smite et al.'s taxonomy (Šmite et al. 2014). For this reason we decided not to consider them in this work. The taxonomy proposed by Narasipuram (2006) was also not considered because its supporting evidence is limited.

The identified knowledge organization schemes can be categorized as follows:

- **Description approach:** Three studies (Laurent et al. 2010; Vizcaíno et al. 2012; Marques et al. 2013) proposed graphical-based approaches that are more adequate to describe rather than to classify GSE projects. It comes from the fact that none of these approaches has dimensions with clear classification criteria associated to them; rather, they provide a set of “variables” that should be instantiated.
- **Classification approach:** The other two studies (Gumm 2006; Šmite et al. 2014) are more adequate to classify GSE projects, because they are organized in dimensions that have categories with associated classification criteria.

2.1 Smite et al.'s GSE taxonomy

Smite et al. (2014) conducted a Delphi-inspired study with GSE researchers to develop an empirically based glossary and taxonomy, focused on the sourcing strategy aspect of GSE projects.

To construct the taxonomy, firstly, the authors investigated the state of the art in the use of GSE terminology by systematically reviewing studies from GSE-related venues. Secondly, by using a Delphi-based approach, they evaluated the literature and defined a consensual terminology. Finally, the authors identified the relationship between the defined terms using the defined terminology. To illustrate the usage of the proposed GSE taxonomy, the authors classified sourcing strategies presented in 68 different studies.

This taxonomy was developed to classify the relationship between pairs of sites, although it is equally possible to describe more complex GSE projects, with more than

two sites. The taxonomy has five dimensions, as presented in Fig. 1 and described in Table 1.

The taxonomy was designed using a facet-based classification structure (Kwasnik 1999). It has five facets (dimensions), which relate to each other as follows:

- The dimension “GSE” is the parent of all the other dimensions.
- The classification by means of the dimension “geographic distance” depends on the categories of the dimension “location.”
- The classification by means of the dimension “temporal distance” depends on the categories of the dimension “geographic distance”.

2.2 Additional related work

Gumm (2006) developed a taxonomy to classify GSE projects in terms of distribution dimensions. Its goal was to provide a foundation to discuss the challenges related to GSE projects and was based on an earlier literature study performed by the same author.

The proposed taxonomy uses four different dimensions (physical distribution, organizational distribution, temporal distribution and distribution between stakeholder groups) to classify the ways in which people and artifacts can be distributed in GSE projects. Each dimension can be measured on a 3-point ordinal scale (high, medium or low). The author describes an onshore distributed project to validate her proposal and she argues that the taxonomy helps to understand the scope and the distribution issues of the evaluated project.

Laurent et al. (2010) proposed a taxonomy and a visual notation to address the requirements engineering aspect of GSE projects. These authors’ main goal was to design a

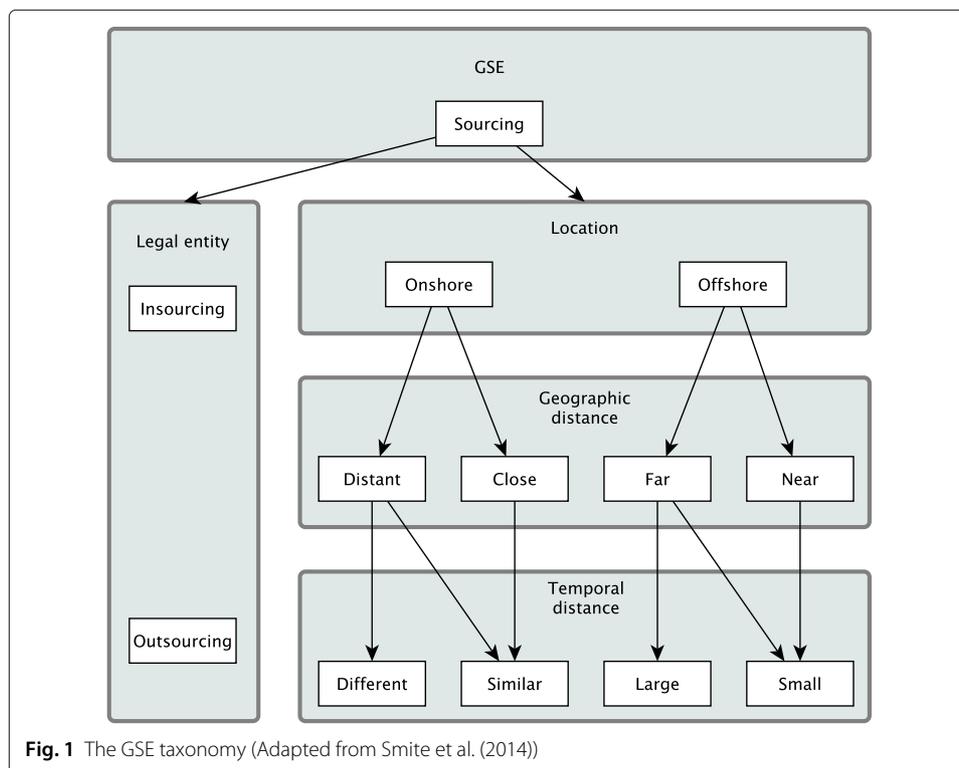


Table 1 Dimensions of Smite et al.'s taxonomy

Dimension	Categories	Description
GSE	Sourcing	This dimension contains the root of the taxonomy, called sourcing. In this context, sourcing means some form of external software development.
Location	Onshore, Offshore	A sourcing can be delegated to a site in the same country, i.e. onshore, or to a site in another country, i.e. offshore.
Legal entity	Insourcing, Outsourcing	Independently from the location, a sourcing can be transferred to a different branch (site) of the company, i.e. insourcing, or subcontracted to a different legal entity (company), i.e. outsourcing.
Geographical distance	Close, Distant, Near, Far	In onshore projects, the geographical distance is considered: close when it is possible to have relatively frequent face-to-face meetings, since no flights are required to go from one site to the other; distant when at least one flight is required to have face-to-face meetings, which yields time and cost increases. In offshore projects, the geographical distance is considered: near when the required flying time is less than two hours; far when the flying time is longer than two hours and staying overnight is usually required
Temporal distance	Similar, Different, Small, Large	In onshore projects, the temporal distance is considered: similar when there is a time difference of one hour or less; different when the time difference between two sites is longer than one hour. In offshore projects, the temporal distance is considered: small when there is a time distance between sites of four hours or less; large when there is a time distance between two sites of more than four hours

common language for modeling the requirements of GSE projects and to allow project managers to manage distributed requirements in a better way. The proposal was derived from the findings of a broad study performed with industrial partners (seven different projects). Interviews were performed with the team leaders responsible for eliciting and gathering the requirements in each project.

The taxonomy is divided into three different entities: role, site and artifact. They graphically showed the taxonomy as a Unified Model Language (UML) class diagram with some attributes in each entity. These attributes are related to the entity "site" and respectively called location, language and time zone.

To facilitate the taxonomy's usage, the authors also designed a visual notation, which was later on used to describe a real GSD project from the video gaming domain. They report that the taxonomy help to identify problems in advance regarding the management of documents and the requirements gathering process.

Vizcaino et al. (2012) developed an ontology, called O-GSD, which was aimed at easing the communication and avoiding misunderstanding in GSE projects. This ontology was iteratively developed in the context of a project that involved five companies and two universities in Spain. The authors used the REFSENO (representation formalism for software engineering) (Tautz and Von Wangenheim 1998) to create the ontology.

The ontology allows for the description of GSE projects by the instantiation of different factors, e.g. time zone difference and language distance, roles of the involved members and involved sites. The authors designed a glossary and a UML class diagram to depict the semantic relationship between all the determined concepts.

To validate their proposal, the ontology was used to describe a real GSE project, which consisted of software related to the sale of security devices in the European Union

countries. The ontology was able to cover all the concepts required by the involved company to represent the GSE project, and in addition it also fostered a common understanding about the represented project.

Marques et al. (2013) introduced an ontology for team task allocation in GSE projects. This ontology was developed based on the findings of a systematic mapping study performed by the authors and aimed at clarifying the concepts related to team task allocation in distributed projects.

The authors used UML class diagrams to represent the ontology. The main concepts addressed by the authors were artifact, competence and constraints. The authors performed a preliminary evaluation of the ontology by interviewing five project managers, and the results suggested that the concepts and relationships embraced by the ontology were suitable to be applied in real distributed projects.

2.3 Research gaps

Only the taxonomies proposed by Gumm (2006) and Smite et al. (base GSE taxonomy) (Šmite et al. 2014) are considered as knowledge classification approaches. The base GSE taxonomy is more comprehensive, providing a wider range of relevant dimensions and clear criteria to classify GSE projects. In addition, this taxonomy was also developed with the participation of several GSE experts, which provides the taxonomy more strength and credibility.

Despite the usefulness of the base taxonomy, to classify GSE projects in a more comprehensive way, additional aspects must be considered (as identified in our previous work (Britto et al. 2014, 2015)):

- It is well-known by the GSE community that language and cultural factors have an important role in GSE projects (Herbsleb and Moitra 2001; Herbsleb and Mockus 2003; Conchúir et al. 2009; Britto et al. 2014; Britto et al. 2015). Despite both factors being discussed by Smite et al. (2014), their taxonomy does not have dimensions to represent these factors.
- The base taxonomy was developed to classify relationships between pairs of sites. Nevertheless, some GSE factors would be better classified in the granularity level of site rather than the granularity level of relationship-between-a-pair-of-sites, e.g. a site's software process type and cultural factors.

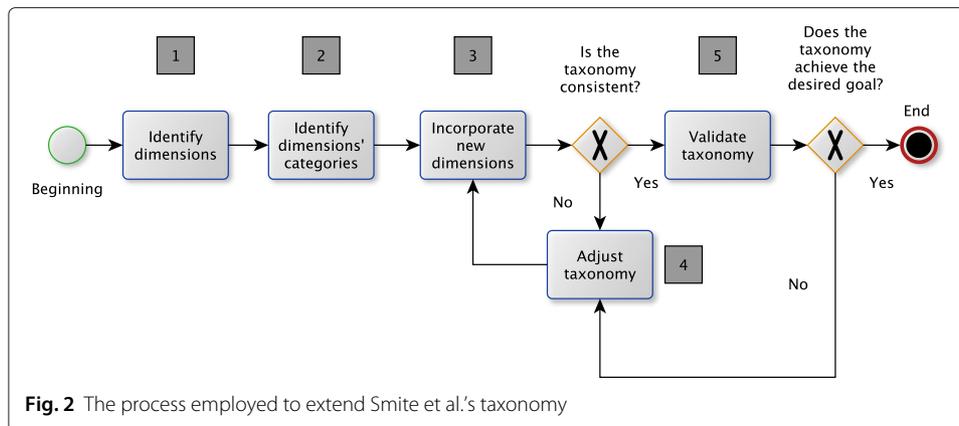
3 Research design and methodology

This section presents the research design and methodology used herein. The following research questions drove the work reported in this paper:

- **RQ1:** What dimensions are needed to augment the usefulness of Smite et al.'s taxonomy?
- **RQ2:** What is the utility of the extended taxonomy?

To answer RQ1 and RQ2, we followed the process presented in Fig. 2.

First, we identified the dimensions to be incorporated into the original taxonomy. The results of one SLR (Britto et al. 2014) and one survey (Britto et al. 2015) were used as input to this step. Only the dimensions reported in more than one primary study in the SLR and later on confirmed by the survey were selected, i.e. only the dimensions with



empirical evidence were considered. In this step, we identified four new dimensions (software process model, cultural factors, language and communication model) not present in the original taxonomy, but judged as essential to capture in the taxonomy.

Second, we identified categories for each dimension. To do so, we used relevant literature related to each dimension (see Section 4) and our own knowledge to identify meaningful categories with clear classification criteria. Clear classification criteria facilitate the usage of the taxonomy, and help in making correct classifications of the subject matter (Wheaton 1968) ¹.

During this step, we identified the need to split the dimensions related to culture and software process, as follows:

- “Culture” was split into two dimensions: “power distance” and “uncertainty avoidance”.
- “Software process” was split into two dimensions: “software process type” and “software process distance”.

We did so to enable our extended taxonomy to classify culture and software process related factors on a finer grained level, which we believe would enhance its usefulness and enable the classification of a wider range of GSE contexts.

Third, we combined the new dimensions with the dimensions of the original taxonomy. In doing so, we identified some inconsistencies in the resulting extended taxonomy; most new dimensions are site-related, but the original taxonomy was designed to classify only relationships between pair of sites (see Fig. 3 in Section 4).

Fourth, to address this inconsistency, we added one new dimension, called “setting”, which enables the classification of GSE projects in both site level and relationship-between-pair-of-sites level. We also adjusted the original dimension “GSE” to keep consistency, i.e. its category was renamed to *project*.

Fifth, we validated our extended taxonomy. A taxonomy can be validated in three ways (Šmite et al. 2014):

- **Orthogonality demonstration** - The orthogonality of the taxonomy dimensions and categories should be demonstrated.
- **Benchmarking** - The taxonomy should be compared with other similar classification schemes.

- **Utility demonstration** - The utility of the taxonomy should be demonstrated through the classification of existing knowledge.

The orthogonality of the new dimensions was ensured by defining categories with clear classification criteria (see Section 4).

The benchmarking was carried out by comparing the extended taxonomy with the base taxonomy (Šmite et al. 2014) and Gumm’s (2006) proposals. We used only these two taxonomies to perform the benchmarking because the other knowledge organization schemes did not provide clear criteria to perform knowledge classification, as discussed in Section 2. Our benchmarking is further detailed in Section 5.2.

Finally, to demonstrate the utility of our extended taxonomy, we illustrate its usage by classifying eight real finished GSE projects. This illustration is presented in Section 5.1.

4 The extended GSE taxonomy

Figure 3 displays the extended GSE taxonomy proposed herein ². Seven new dimensions were incorporated into the base taxonomy: “setting”, “software process type”, “software process distance”, “power distance”, “uncertainty avoidance”, “language distance” and “communication model”.

The relationship between the new and original dimensions are as follows:

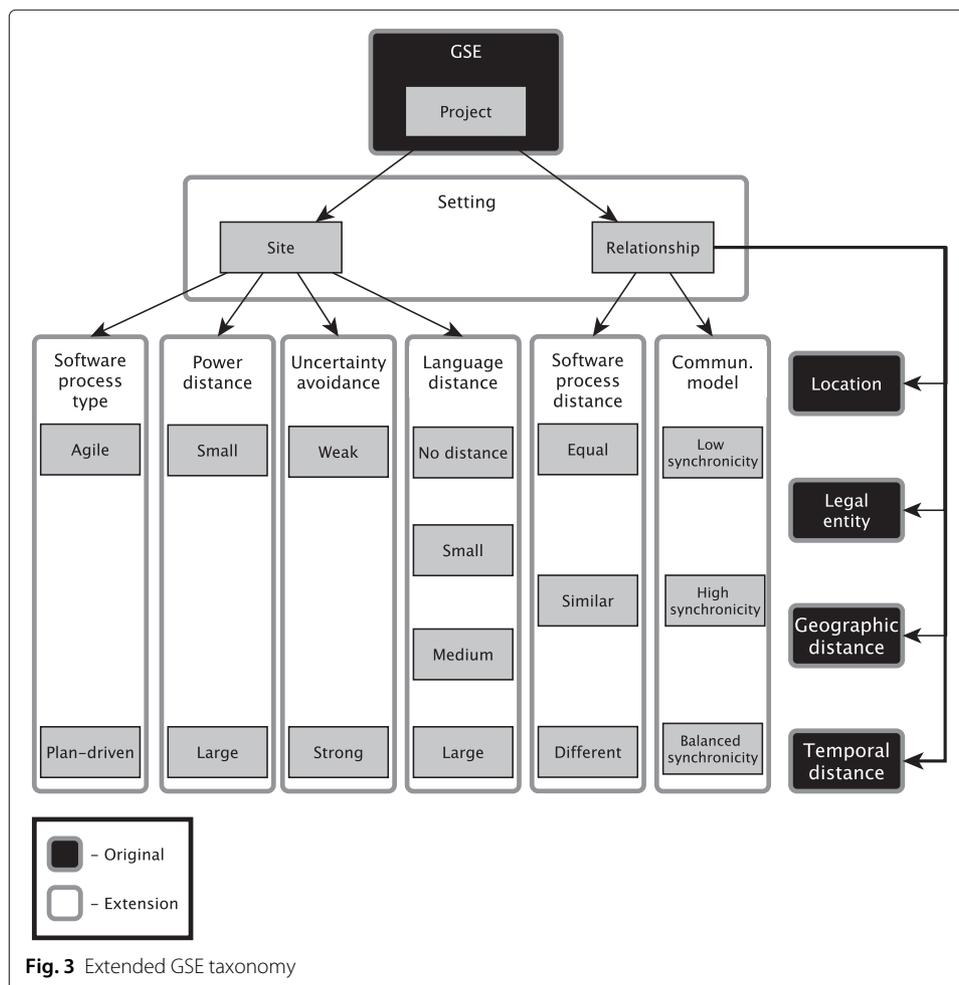


Fig. 3 Extended GSE taxonomy

- The dimension “GSE is the parent of all the other dimensions.
- The classifications by means of the dimensions “software process type, “power distance, “uncertainty avoidance and “language distance are related to the category *site* of the dimension “setting.
- Classifications by means of the dimensions “software process distance and “communication model are related to the category *relationship* of the dimension “setting.

The seven new dimension are further detailed in Sections 4.1 (GSE), 4.2 (setting), 4.3 (software process type and software process distance), 4.4 (power distance and uncertainty avoidance), 4.5 (language distance) and 4.6 (communication model) respectively.

4.1 GSE

GSE is the root of the taxonomy. To better manage to classify on both site and relationship-between-pair-of-sites granularity levels, *project* has been introduced into the root level instead of *sourcing*, as proposed in the original taxonomy. Herein we consider a project as “a temporary endeavor undertaken to create a unique product, service, or result (Institute 2013).

4.2 Setting

The dimensions of the extended taxonomy are formulated to classify GSE projects on both the site level (Site) and the relationship-between-pair-of-sites level (Relationship).

A *site* is defined as a unit composed of human resources that interact with other sites (nodes). We define a *relationship* as the relationship between two sites interacting in a project (edge).

4.3 Software process dimensions

The software development process type used (agile Royce (1987), plan-driven Sommerville (2010) or hybrid Kuhrmann and Mendez Fernandez (2015); Vijayarathy and Butler (2015)) is an aspect that impacts the conduct of a GSE project, e.g. the effort required to perform such projects (Britto et al. 2014, 2015). In addition, the way the practices are incorporated into the sites’ routines can also be different (workflows). Differences between software processes used in different sites may lead to problems in the communication and loss of trust (Ramasubbu et al. 2011), for example impacting the associated effort (Britto et al. 2014, 2015).

Therefore, we incorporated the dimensions *software process type* and *software process distance* to account for software process factors.

4.3.1 Software process type

Plan-driven software development may be viewed as heavy and bureaucratic to deal with certain types of projects, specially the ones where the requirements are unclear and uncertain (Fernandez and Fernandez 2008). Therefore, the main criticism regarding plan-driven development is that many decisions that are taken early on must be reappraised later on, since software development deals with a lot of uncertainty in the early stages of a project (Pfleeger 1999). Nevertheless, this approach allows for planning organizational aspects earlier, besides fostering the discovery of potential problems before the start of a particular project.

Agile methods are regarded as being more suitable to deal with projects that present unclear and uncertain requirements, but they demand close collaboration between the customer and the development team (Beck and Andres 2004). Furthermore, organizations and customers may be more familiar with plan-driven approaches and may find it hard to trust and follow an agile-based approach (Gandomani and Zulzalil 2013). Pure agile-based software processes are difficult to scale. They are more adequate to small and medium size projects (Gandomani and Zulzalil 2013). Finally, existing empirical evidence suggests that agile practices are not readily applicable to GSE projects (Hossain et al. 2009; Jalali and Wohlin 2012).

A software process at the project level may be split, which enables distributed teams to combine practices from both agile and plan-driven approaches, and hence generating software process diversity (Ramasubbu et al. 2015). Software diversity can help teams to address the limitations of pure agile or plan-driven software processes by combining the practices from each approach that fits each case (Ramasubbu et al. 2015), thus leading to hybrid processes (Kuhrmann and Mendez Fernandez 2015; Vijayarathy and Butler 2015). For example, some organizations have a more plan-driven mentality about project management practices, but the teams may still be mainly agile.

Considering the discussion above, we define the dimension “software process type as having two categories:

- **Agile** - The software process used in a site is mainly based on agile practices.
- **Plan-driven** - The software process used in a site is mainly based on plan-driven practices.

We did not include a category *hybrid* in this dimension, because one of the types of practice (agile or plan-driven) is expected to be more prevalent. Furthermore, most organizations would probably perceive themselves as using a hybrid approach, and hence it is viewed as more important to know the process type being most commonly used with respect to the objective of a study. For example, an organization may have agile teams that are managed in a more plan-driven way; if the main focus of the classification is the effort associated with the software development in each site, the best classification would be *agile*, because the teams use mainly agile practices to develop software; however, if the main focus of the classification is the effort associated with coordination between sites, *plan-driven* would fit the best, since management is more plan-driven than agile.

4.3.2 *Software process distance*

While software diversity can help teams to overcome the limitations associated with “pure software processes (pure agile or pure plan-driven), it may result in differences between the software processes of different sites. To account for this, we incorporated the dimension *software process distance*, which enables the classification of the distance between two sites in terms of the software processes used. This dimension has the following categories:

- **Equal** - The software processes of the sites are very similar, i.e. they use the same workflows, roles and practices to develop software.
- **Similar** - The workflows, roles and practices are not the same in both sites, but the sites use software processes that are based on the same type of software development practices (mainly agile or mainly plan-driven).

- **Different** - The sites neither use the same type of software development practices nor use the same workflows, roles and practices.

4.4 Cultural factors

Both national and organizational cultures influence both decision-making and the way development is conducted in a project. In GSE projects, the different cultures involved can impact negatively on the communication and trust between sites (Da Silva et al. 2010), and can lead, for example, to a bigger effort (Britto et al. 2014, 2015).

Culture is represented in our extended taxonomy using Hofstede's national culture framework (Hofstede et al. 2010). From Hofstede's framework, we have only adopted two dimensions that are named *power distance index - PDI* and *uncertainty avoidance index - UAI*. They have been adopted because empirical evidence exists for these two dimension; the evidence supports their influence on the organizational level (Hofstede et al. 2010), which is the level in which projects are carried out.

Hofstede's PDI and UAI are defined as follows:

- **Power distance index (PDI)** - Measures how people manage inequality in hierarchical relationships, i.e. manager-subordinates. In nations with high PDI, the employees depend more on the managers to make decisions. However, in nations with low PDI, the competences of the employees are higher valued than their hierarchical position.
- **Uncertainty avoidance index (UAI)** - Measures how people manage uncertainty, how they feel threatened by uncertain situations and try to avoid or mitigate such situations. In nations with strong uncertainty avoidance, they have strict laws and rules. Nevertheless, nations with weak uncertainty avoidance have as few rules as possible, which make their people more tolerant to uncertain situations.

Based on Hofstede's framework, we designed the dimensions *power distance* and *uncertainty avoidance* to account for the cultural factors in our extended taxonomy.

4.4.1 Power distance

PDI is represented in our extended taxonomy by the dimension called *power distance* (PD), which has the following categories:

- **Small** - Sites placed in countries with $PDI \leq 50$ have small power distance.
- **Large** - Sites placed in countries with $PDI > 50$ have large power distance.

4.4.2 Uncertainty avoidance

In our extended taxonomy, UAI is represented by the dimension called "uncertainty avoidance (UA), which has the following categories:

- **Weak** - Sites placed in countries with $UAI \leq 63$ have weak uncertainty avoidance.
- **Strong** - Sites placed in countries with $UAI > 63$ have strong uncertainty avoidance.

The threshold values used to differentiate sites with *Small* or *Large* PD and *Weak* or *Strong* UA were defined based on Hofstede et al.'s empirical study (Hofstede et al. 2010)³. To choose the proper UA and PD categories for a site, UAI and PDI scores for the countries involved in a project under classification should be determined; the scores are available in Hofstede et al.'s book (Hofstede et al. 2010).

Note that outcomes of the PD and UA classifications for sites should be compared. For example, consider a project with two sites, respectively named X and Y. Site X is placed in Germany and site Y is placed in Brazil, i.e. X's PD is *Small* and UA is *Strong*, while Y's PD is *Large* and UA is *Strong*. In this example, there is no major concern about the impact of UA on the GSE project, since both sites are classified in the same category. However, PD could negatively impact the GSE project, since the sites are classified in different categories.

In some situations, companies source human resources from different countries to compose teams in one location. This means that the main national culture of a particular site is not necessarily the national culture of the country wherein the site is placed. For example, Ramasubbu and Balan (2007) report a project with two sites, one placed in the USA and the other one located in India. Although both sites are placed in countries with different PD and UA, the human resources of both sites were from India. In such situation, the actual cultural distance between the two sites is expected to be zero.

Therefore, it is important to account for the predominant nationality of the human resources of a site to define the appropriate PD and UA.

4.5 Language distance

In a GSE project, it is very likely that involved sites do not have the same native language, which may lead to misunderstandings between sites and generate delays in the entire project (Ågerfalk et al. 2005). Nowadays, English is the most commonly used language when there is need for a *lingua franca* (Lutz 2009). Thus, instead of calculating the distance between sites' languages, we incorporated a dimension named *language distance*, which classifies the distance between each site's language and English.

This dimension has the following categories:

- **No distance** - When the mother language of a site is English, or no *lingua franca* is required. In the latter case there is no language distance in such a site, since people from both sites could communicate in their native tongue.
- **Small** - When $0 < Ld \leq 0.4$, the language distance of a site is considered small. This means that it is more likely that people from such a site have an acceptable level of proficiency in English, since it is relatively easy for them to learn it.
- **Medium** - When $0.4 < Ld \leq 0.57$, the language distance of a site is considered medium. This means that it is more likely that people from such a site struggle somewhat to learn English, which affects their proficiency. However, they can learn and speak English by applying more effort than people from the previous group
- **Large** - When $0.57 < Ld \leq 1$, the language distance of a site is considered large. This means that it is more likely that people from such a site struggle even more to learn English. In general, those languages have almost no commonalities with English, which requires more effort to learn English.

In the aforementioned categories, Ld represents the distance between the language of a particular site and English (Chiswick and Miller 2004). According to Chiswick et al., Ld can assume the following values: 0.33, 0.36, 0.4, 0.44, 0.5, 0.57, 0.67, 0.8 and 1⁴.

The bigger the Ld value, the farther a particular language L is from English, which is also a measure of how difficult it is for people who speak L to learn to speak English. Thus, the larger the Ld , the higher the likelihood that the proficiency in English will not be very good (Chiswick and Miller 2004). The lower the level of proficiency in English

(as *lingua franca*), the higher the probability of problems regarding the communication between sites (Herbsleb and Moitra 2001).

Note that this dimension of our taxonomy can be used only in projects that require no *lingua franca* to enable communication between sites (i.e. there is no language distance) or when the chosen *lingua franca* is English.

The first category of this dimension (*No distance*) was designed to represent sites that either have English as its mother tongue or no *lingua franca* is required in the project, since the sites have the same mother tongue. The other three categories were defined by dividing the language distance scale in three equal parts (*Small*, *Medium* and *Large*), so that there is enough representativeness to classify the existing spectrum of language distance values.

This dimension focuses on the language that is spoken the most in a site's location. We did so because it would be very difficult to embrace the particularities of countries that have more than one official language. In addition, high proficiency in English is a prerequisite to allocate personnel to participate in many GSE projects, i.e. in these cases, the mother tongues of sites' locations are not an issue.

Thus, when using this dimension to classify the language distance of each site, the language spoken the most by the site's personnel should be identified and it should be used as basis for selecting the language distance category that fit the best.

4.6 Communication model

In GSE projects, the communication between the distributed sites is often mediated via electronic communication media (Jaanu et al. 2012) and existing empirical evidence shows that mediated communication demands more effort (Ebert and De Neve 2001; DeLuca and Valacich 2005). Different electronic communication media types have different properties and capabilities to deal with geographic distance between sites in GSE projects.

Media synchronicity theory (MST) (Dennis et al. 2008) states that the most effective communication occurs when the communication media matches a given set of communication requirements (Dennis et al. 2008). The information to be transmitted can require more conveyance, i.e. processing and transmission of new information, or more convergence, i.e. a group should agree on something (Dennis et al. 2008).

In our extended taxonomy, we incorporated the communication factor through the dimension called *communication model*, which is based on the MST. The communication model dimension has the following categories:

- **Low synchronicity communication model** - The communication between sites is mainly mediated via asynchronous media, e.g. email and issue trackers (the most adequate media type for conveyance (Dennis et al. 2008)).
- **High synchronicity communication model** - The communication between sites is mainly mediated via synchronous media, e.g. video conference and instant messaging tools (the most adequate media type to achieve consensus (Dennis et al. 2008)).
- **Balanced synchronicity communication model** - The communication model encompasses both asynchronous and synchronous media types and each type is used for its most adequate purpose, i.e. synchronous media is used when there is need for fast feedback and consensus achievement, and asynchronous media is used when

there is need to convey some message that should be formalized and consolidated before transmission.

5 Validation

A taxonomy can be validated through orthogonality demonstration, benchmarking and utility demonstration. We ensured the orthogonality of the extension's dimensions by defining categories with clear classification criteria (see Section 4). We demonstrate the utility of the extension in Section 5.1 and a benchmarking is presented in Section 5.2.

5.1 Demonstration of utility

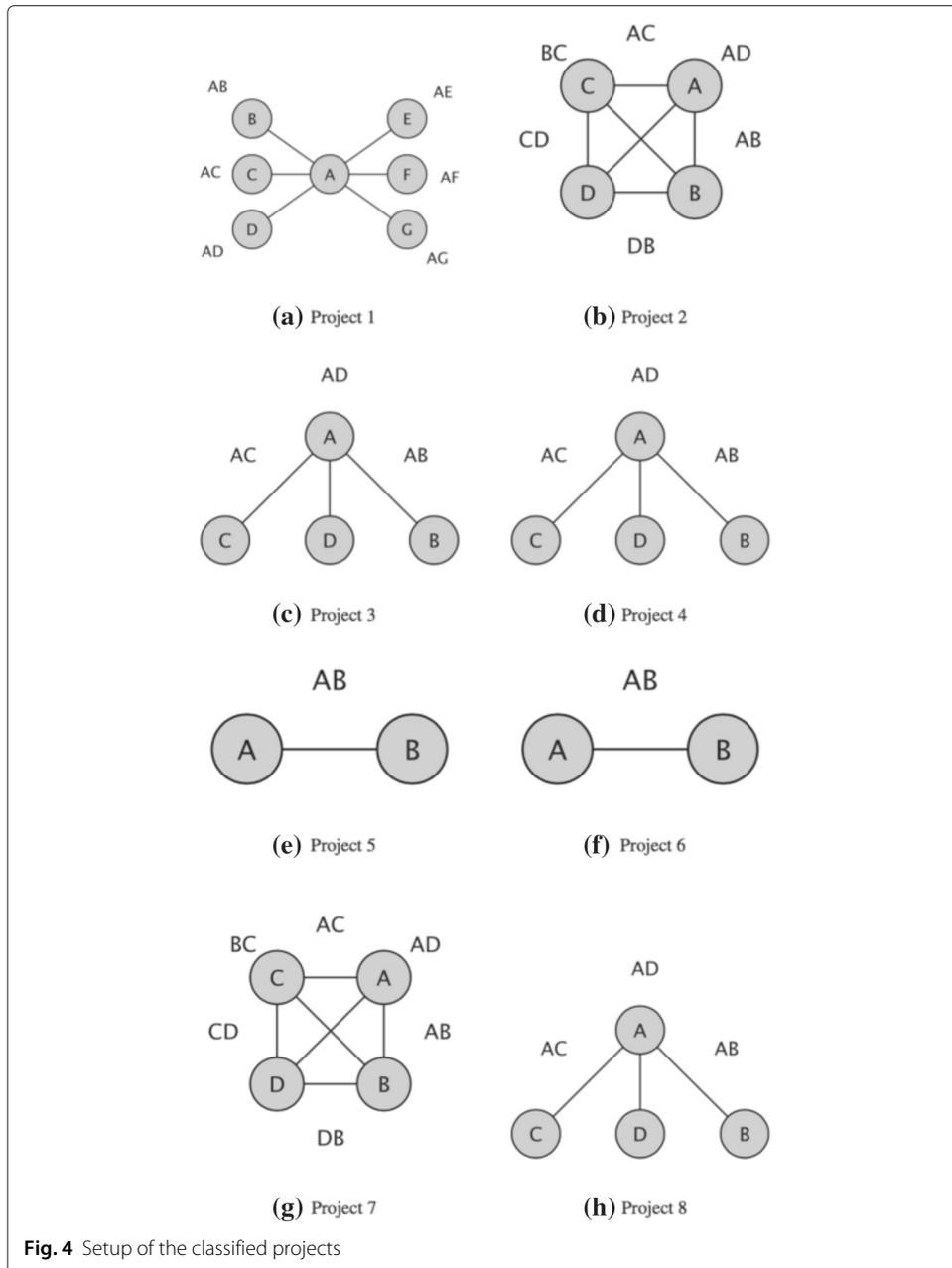
To demonstrate the utility of our extended taxonomy, we classified eight finished real GSE projects. These projects were obtained from Ramasubbu et al. (2012). Since not all the projects' data required to perform the classification was available in the paper, we contacted the authors to complement the missing data. We provided them an excel spreadsheet with eight tabs (one per project). Each tab had the following fields: project's ID (PID), company's name, software domain, site's ID (SID), site's country, site's city, site's main language, software process type, software process distance, communication model, legal entity, relationship ID (RID) and relationship's sites. Clarifications related to the spreadsheet was provided whenever required.

Due to a non-disclosure agreement, the name of the companies and the domain of the software applications developed/maintained were not provided. The cities wherein the sites were placed were also not explicitly stated, also because of the non-disclosure agreement. Rather, they presented the location of the sites in terms of countries. In the case of sites placed in the USA, the region (e.g. north, south, east, west) of each site was also provided. In addition, the authors also clarified that the team members of all projects were fluent in English. Thus, we considered that there was no language distance between the sites of the projects.

We used the provided data to conduct the classification of the eight projects. Figure 4 shows the setup of each project, i.e. the connections between the involved sites⁵, while Table 2⁶ shows the classification for each project in the site level and Table 3 shows the classification of each project in the relationship-between-pair-of-sites level.

The projects classified by means of the proposed extended dimension provided a variety of different setups:

- Project 1 had seven sites and project management was concentrated in site A (USA), i.e. there was no interaction between sites B, C, D, E, F and G; all the sites only interact with A.
- In Project 2, four onshore sites (USA) were involved and all the sites directly interact with each other, although site A had the biggest responsibility regarding project management.
- Project 3 had four sites and project management was concentrated in site A (India). Despite the fact that sites B, C and D were all located in the USA, they did not interact with each other, only with site A.
- In Project 4, four sites were involved and project management was concentrated in site A (India). The other sites only interact with site A.



- Project 5 and project 6 had two sites involved each and project management was mainly the responsibility of site A for both project 5 and project 6 (India and Germany respectively).
- In Project 7, four onshore sites (India) were involved and all the sites directly interact with each other, although site A had the main responsibility regarding project management.
- Project 8 had four sites and project management was concentrated in site A (USA). Despite the fact that sites B, C and D were all located in India, they did not interact with each other, only with site A.

The classification results can be used in many different ways, such as:

Table 2 Classification in the site level

PID	SID	Country	Software process type	Power distance	Uncertainty avoidance
1	A	USA	Agile	Small	Weak
1	B	USA	Agile	Small	Weak
1	C	India	Plan-driven	Large	Weak
1	D	India	Plan-driven	Large	Weak
1	E	Singapore	Plan-driven	Large	Weak
1	F	Germany	Agile	Small	Strong
1	G	Australia	Agile	Small	Weak
2	A	USA	Agile	Small	Weak
2	B	USA	Agile	Small	Weak
2	C	USA	Agile	Small	Weak
2	D	USA	Agile	Small	Weak
3	A	India	Plan-driven	Large	Weak
3	B	USA	Agile	Small	Weak
3	C	USA	Agile	Small	Weak
3	D	USA	Agile	Small	Weak
4	A	India	Plan-driven	Large	Weak
4	B	Germany	Agile	Small	Strong
4	C	Spain	Agile	Small	Strong
4	D	England	Agile	Small	Weak
5	A	India	Plan-driven	Large	Weak
5	B	Japan	Plan-driven	Large	Strong
6	A	Germany	Agile	Small	Strong
6	B	India	Agile	Large	Weak
7	A	India	Plan-driven	Large	Weak
7	B	India	Plan-driven	Large	Weak
7	C	India	Plan-driven	Large	Weak
7	D	India	Plan-driven	Large	Weak
8	A	USA	Agile	Small	Weak
8	B	India	Plan-driven	Large	Weak
8	C	India	Plan-driven	Large	Weak
8	D	India	Plan-driven	Large	Weak

- To help researchers to classify studies so that other researchers and practitioners can more easily identify cases of particular interest to them. This directly relates to the next item.
- To identify cases of interest in the literature; the extended taxonomy provides a richer classification than the existing taxonomies. Thus, it improves the identification of similar cases in the literature. For example, if a practitioner or researcher intends to find literature related to GSE projects with sites that use different software processes, it would be easier to identify relevant studies if researchers report the studies using the extended taxonomy.
- To identify relationships between the classification results associated with each dimension and some other factor, which eventually can support decision-making processes. For example, it would be possible to use the classification results along with the effort associated with software development projects to perform a regression

Table 3 Classification in the relationship-between-pair-of-sites level

PID	RID	Process	Communication	Location	Legal entity	Geographic	Temporal
1	AB	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Different
1	AC	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
1	AD	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
1	AE	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
1	AF	Similar	Balanced synchronicity	Offshore	Insourcing	Far	Large
1	AG	Similar	Balanced synchronicity	Offshore	Insourcing	Far	Large
2	AB	Similar	Balanced synchronicity	Onshore	Outsourcing	Distant	Similar
2	AC	Similar	Balanced synchronicity	Onshore	Outsourcing	Distant	Different
2	AD	Similar	Balanced synchronicity	Onshore	Outsourcing	Distant	Different
2	BC	Similar	Balanced synchronicity	Onshore	Outsourcing	Distant	Similar
2	BD	Similar	Balanced synchronicity	Onshore	Outsourcing	Distant	Different
2	CD	Similar	Balanced synchronicity	Onshore	Outsourcing	Distant	Similar
3	AB	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
3	AC	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
3	AD	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
4	AB	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
4	AC	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
4	AD	Different	Balanced synchronicity	Offshore	Insourcing	Far	Large
5	AB	Similar	Balanced synchronicity	Offshore	Insourcing	Far	Large
6	AB	Similar	Balanced synchronicity	Offshore	Insourcing	Far	Large
7	AB	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Similar
7	AC	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Similar
7	AD	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Similar
7	BC	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Similar
7	BD	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Similar
7	CD	Similar	Balanced synchronicity	Onshore	Insourcing	Distant	Similar
8	AB	Different	Balanced synchronicity	Offshore	Outsourcing	Far	Large
8	AC	Different	Balanced synchronicity	Offshore	Outsourcing	Far	Large
8	AD	Different	Balanced synchronicity	Offshore	Outsourcing	Far	Large

analysis and identify the relationship between the factors in the classification and the effort of GSE projects. The regression analysis result could be used to define which setup would be more adequate for a particular situation. The extended taxonomy allows for identifying relationships that involves software process-related aspects, socio-cultural/language aspects, communication aspects and site-related aspects.

5.2 Benchmarking

To further validate and justify the need for the extended taxonomy presented herein, we compared the extension with the base taxonomy (Šmite et al. 2014) and Gumm's (2006) taxonomy. We performed the comparison considering the following aspects:

- The classification results presented in Section 5.1.
- The taxonomy's basis.
- Type of classification structure used to design the taxonomy, which can be hierarchy, tree, paradigm and facet-based (Kwasnik 1999).

- Procedure used to classify GSE projects, which can be qualitative or quantitative (Wheaton 1968).
- Description of the dimensions' categories (classification criteria), which can be objectively or subjectively described.
- Validation approach.

Table 4 displays the values related to the aforementioned factors, which are further elaborated next.

5.2.1 Classification results

Considering the base taxonomy's lack of certain dimensions as described in Section 2, including not covering software process-related aspects, socio-cultural/language-related aspects, communication-related aspects and site-level aspects, the extended taxonomy presented herein adds new dimensions that are believed to be important to better understand the context of GSE projects. Thus, our extended taxonomy can simplify the comparison between different studies, in particular when classifying new studies, so making it easier for both researchers and practitioners to compare their new cases with the existing literature on the topic.

When analyzing the classification results presented in Tables 2 and 3, the base taxonomy classification coverage is equal to 42 % (lack of seven dimensions) when compared to the extended taxonomy, while Gumm's taxonomy is able to cover only 25 % (lack of nine dimensions). This means that the dimensions that are relevant to describe GSE research contexts in a more comprehensive way are only covered by the extended taxonomy. In summary, the base taxonomy and the taxonomy by Gumm are subsets of the extended taxonomy.

5.2.2 Basis

The base taxonomy is an empirically based taxonomy, i.e. it was designed based on existing literature and also on the knowledge of several experts. Our extended taxonomy is based on the base taxonomy, two systematic empirical studies and expert knowledge. Gumm's taxonomy was based on a non-systematic literature review. Thus, Gumm's taxonomy lacks the empirical basis as compared to our extended taxonomy and the base taxonomy.

5.2.3 Classification structure

All three taxonomies were designed using faceted analysis (Kwasnik 1999) (facet-based classification structure), i.e. that GSE projects are classified along different perspectives (dimensions). It is not surprising that the three taxonomies were designed via this approach, because faceted analysis is the most adequate approach to classify knowledge of new and evolving knowledge areas (Kwasnik 1999), which is the case of GSE.

5.2.4 Classification procedure

Classification procedures define how subject matter instances (GSE projects) are systematically assigned to the categories of each dimension (Wheaton 1968). Qualitative classification procedures are based on scales while quantitative classification procedures are based on ratio scales (Wheaton 1968). The three taxonomies use nominal scales.

Table 4 Comparison between the extended taxonomy, the base taxonomy and Gumm’s taxonomy

Taxonomy	Basis	Dimensions	Structure	Procedure	Description	Validation
Extended	Existing taxonomy, Systematic literature review and survey findings, and expert knowledge	12 (all the identified dimensions)	Facet-based	Qualitative	Objectively described	Comparison with existing taxonomies and demonstration of utility using data from eight finished GSE projects.
Base	Existing taxonomies, literature review and expert knowledge	5 (GSE, location, legal entity, geographic distance and temporal distance)	Facet-based	Qualitative	Objectively described	Based on expert judgment, comparison with existing taxonomies and demonstration of utility by classifying GSE literature.
Gumm’s	literature review	3 (legal entity, geographic distance and temporal distance)	Facet-based	Qualitative	Subjectively described	Demonstration of utility using data from one finished GSE project.

5.2.5 Classification criteria description

Both our extension and the base taxonomy provide very clear and objective descriptions for all their dimensions' categories, i.e. the classification criteria are clear. Gumm does not provide a very clear description of its taxonomy's categories, which makes it harder to use her taxonomy when compared to the usage of the other two taxonomies.

5.2.6 Validation

As discussed in Section 3, a taxonomy can be validated through orthogonality demonstration, benchmarking and utility demonstration.

The dimensions' orthogonality in both our proposal and the base taxonomy was ensured by defining categories with clear classification criteria (see Section 4). Gumm does not discuss or demonstrate the orthogonality of her taxonomy's dimensions.

The benchmarking was done by comparing the extended taxonomy with the two other taxonomies, i.e. Smite et al.'s and Gumm's taxonomies respectively.

The utility of each one of the three proposals was demonstrated by using each taxonomy to perform actual classification. We did so by classifying eight real finished GSE projects whose data was provided by a GSE expert. Smite et al. classified the sourcing strategy of projects reported in 296 research papers. They also provided an example to explain how the taxonomy can be used to identify literature related to specific settings. Gumm used her taxonomy to classify one finished real GSE project.

In our case, an expert provided all the data required to perform the classification. In Smite et al.'s case, they extracted the required information from the classified papers. In some cases they had to infer the data from the text, since the required data was not clear up front; however, this was intentional to illustrate how the data in many papers is insufficient to classify studies of GSE. Gumm does not explicitly explain how the required data to perform the classification was obtained.

6 Discussion

Looking at the gaps (absent dimensions) associated with the base taxonomy, the existing literature suggests their relevance, and hence supports the need for a classification scheme that addresses them. This allows both researchers and practitioners to identify the most relevant cases in the literature in relation to their respective contexts. Abufardeh and Magel (2010) showed that the research addressing the impact of the cultural and linguistic aspects of software developed in a global distributed manner is limited. Mishra and Mishra (2014) have conducted a literature review on cultural issues in GSE. They noted that the findings in the primary studies were not reported in a standard way. Ramasubbu et al. (2015) have recently brought up the issue of software diversity, which has a significant impact on globally distributed projects and is to be yet further investigated. Jaanu et al. (2012) have highlighted the importance of the type of media for effective communication in GSE projects, although GSE researchers are not reporting this aspect very often.

Our extended taxonomy can draw the attention of GSE researchers about the aforementioned aspects and help them to report these aspects in a more comparable way. It has also the potential to foster new research, specifically for the areas that are less covered by the existing literature (e.g. software process diversity.)

To support the research community to “speak the same language, it is important to ensure that there is a consensual terminology and that this consensual terminology will not fragment over time. This work illustrates that it is possible to keep evolving a taxonomy of general use without fragmenting its content. It is important to emphasize that the extended taxonomy is fully consistent with the base taxonomy and can be used to classify GSE projects in any type of GSE study. Although the need for an extended taxonomy came from two studies related to effort estimation in GSE (Britto et al. 2014, 2015), its usage is not limited to effort-related studies.

The extended taxonomy can help GSE researchers to report the context of new GSE research in a more systematic, clear and comparable way. Therefore, it can facilitate the analysis, comparison and aggregation of results from new studies, fostering advances in the knowledge field. Once new studies are reported in a more standardized manner, researchers will also be able to spend less effort to find relevant literature whenever necessary. The use of the extended taxonomy to report GSE studies can help practitioners to identify useful literature related to different contexts and consequently also helping them to address different problems whenever required.

In summary, the extended taxonomy allows for the aforementioned benefits by complementing existing taxonomies when it comes to the classification of GSE projects accounting for software process, socio-cultural, language and communication related factors, in both site and relationship-between-pair-of-sites granularity levels.

7 Limitations

As with most studies, the research reported herein comes with some limitations. First, the dimensions of our extended taxonomy, both the new ones based on the two empirical studies by Britto et al. (2014, 2015) and the original dimensions by Šmite et al. (2014), do not represent an exhaustive list. However, the taxonomy can be further extended new factors are identified. Furthermore, it can be specialized by the incorporation of factors that are of interest for only a particular perspective (e.g. effort estimation). Thus, we encourage other GSE researchers to look at GSE projects from other perspectives (e.g. coordination, testing and effort estimation) to identify potential new dimensions that can augment the representativeness of our proposal.

The extended taxonomy has not yet been used by practitioners in ongoing projects, i.e. there is no empirical evidence ensuring the utility of our proposal in this kind of situation. So, we also encourage GSE researchers to conduct studies in industrial settings to strengthen the usefulness of our proposal for the software-intensive industry.

8 Conclusions

This paper presents an extended taxonomy for classifying GSE projects, which is based on a previous taxonomy (Šmite et al. 2014), two empirical studies (Britto et al. 2014, 2015) and on expert knowledge.

We addressed the first research question of this study (RQ1) by incorporating seven new dimensions into Šmite et al.’s taxonomy, named “setting, “software process type, “software process distance, “power distance, “uncertainty avoidance, “language distance and “communication model.

To validate the extended taxonomy and demonstrate its utility (RQ2), we benchmarked our proposal to two other taxonomies (Šmite et al.’s taxonomy and Gumm’s taxonomy)

and we also illustrate the usage of our taxonomy by classifying eight real finished GSE projects.

The results show that the extended taxonomy can help both researchers and practitioners by facilitating the reporting and understanding of GSE research. Although the new dimensions presented herein were identified in two studies related to effort estimation, the extended taxonomy presented in this paper can be used to report any type of GSE study.

The list of dimensions in our extended taxonomy does not represent an exhaustive list of relevant GSE-related dimensions. Therefore, we intend to conduct further investigation to identify other dimensions that could be incorporated into the taxonomy, so that GSE projects could be classified in a more comprehensive way. More specifically, we intend to identify dimensions related to the way effort estimation processes are framed in GSE projects and how these factors relate to the effort of GSE projects.

Endnotes

- ¹ In this paper, the subject matter to be classified is a GSE project.
- ² The details of the original dimensions are not shown to facilitate the presentation of the new dimensions. Figure 1 illustrates the original dimensions.
- ³ Hofstede et al.'s (2010) provide a table and a figure that contains the UAI and PDI of many different countries.
- ⁴ Chiswick et al. (2004) provide a table that contains the Ld of many different countries.
- ⁵ The nodes represent the sites and the edges represent the relationships between sites.
- ⁶ Due to the fact that there was no language distance in any of the projects, we omitted this dimension in Table 2.

Abbreviations

SE: software engineering; GSE: Global software engineering; SLR: systematic literature review; *Ld*: language distance; PDI: Power distance index; UAI: uncertainty avoidance index; PD: power distance; UA: uncertainty avoidance; PID: project ID; SID: site ID; RID: relationship ID.

Acknowledgments

We would like to thank CNPq (National Counsel of Technological and Scientific Development - Brazil), UFPI (Federal University of Piauí - Brazil) and INES (National Institute of Science and Technology for Software Engineering - Brazil) for partially supporting this work. Furthermore, we would like to thank The Knowledge Foundation, Sweden for partially funding the research through the TEDD (Technical Excellence in Distributed Development) project (grant no. 20120200). We also would like to thank Dr. Narayan Ramasubbu for helping us with the validation of the proposal put forward in this paper.

Authors' contributions

This paper is the result of a joint effort of the three authors. The ideas and solutions put forward are the results of several meetings, in which all the authors had active participation either providing or refining ideas. The research questions and structure of this paper were also defined jointly by the three authors. Besides the aforementioned contributions, the individual contributions of each author of this paper are as follows: RB - He was responsible for identifying related work (Section 2), designing and executing the process to extend Smite et al.'s taxonomy (Section 3), designing the categories of the new dimensions (Section 4), executing the validation (Section 5) and being the main author of the manuscript. CW - He conceptualized the main idea of the paper, i.e. to extend Smite et al.'s taxonomy using the results of Britto et al.'s SLR as a basis. He also provided expert knowledge on GSE and knowledge about the original taxonomy. Finally, he also contributed with feedback and editorial work on the manuscript. EM - She provided expert knowledge on effort estimation and contributed with feedback and editorial work on the manuscript. All authors read and approved the final manuscript.

Authors' information

Ricardo Britto is a PhD student of the Department of Software Engineering at Blekinge Institute of Technology, Sweden. He is also a lecturer of the Department of Computing at Federal University of Piauí, Brazil. Britto received an MSc in Electrical Engineering from Federal University of Rio Grande do Norte in 2008. His research interests include empirical research and software process improvement. Contact him at ricardo.britto@bth.se or visit his website at lattes.cnpq.br/3165532253485275.

Claes Wohlin is a professor of software engineering and dean of the Faculty of Computing at Blekinge Institute of Technology, Sweden. He has previously held professor chairs at the universities in Lund and Linköping. Wohlin received a PhD in Communication Systems from Lund University (SE) in 1991. His research interests include empirical research and software process improvement. He has been Co-Editor-in-Chief and Editor-in-Chief of Information and Software

Technology for 15 years. Claes Wohlin was the recipient of Telenor's Nordic Research Prize in 2004. He is a member of the Royal Swedish Academy of Engineering Sciences since 2011. Contact him at claes.wohlin@bth.se or visit his website at www.wohlin.eu.

Emilia Mendes is a professor in software engineering at Blekinge Institute of Technology, Sweden, and also a Finnish distinguished professor at the University of Oulu, Finland. She has previously been an associate professor at universities in Auckland (NZ) and Zayed (UAE). Mendes received a PhD in Computer Science from the University of Southampton (UK) in 1999. Her research interests include computer science, empirical software engineering and web engineering. To date she has published over 200 refereed publications, which include three books. She worked in the ICT industry for ten years as programmer, business analyst and project manager prior to moving to the UK in the end of 1995 to initiate her PhD studies. Contact her at emilia.mendes@bth.se.

Competing interests

The authors declare that they have no competing interests.

Received: 21 July 2015 Accepted: 12 May 2016

Published online: 07 June 2016

References

- Abufardeh S, Magel K (2010) The Impact of Global Software Cultural and Linguistic Aspects on Global Software Development Process (GSD): Issues and Challenges. In: Proceedings of 4th International Conference on New Trends in Information Science and Service Science - NISS'10, Gyeongju, Korea. pp 133–138
- Ågerfalk P, Fitzgerald B (2008) Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy. *MIS Q* 32(2):385–409
- Ågerfalk PJ, Fitzgerald B, Holmström H, Lings B, Lundell B, Conchúir EO (2005) A framework for considering opportunities and threats in distributed software development. In: Proceedings of the International Workshop on Distributed Software Development. pp 47–61
- Beck K, Andres C (2004) *Extreme Programming Explained: Embrace Change*. 2nd edn. Addison-Wesley
- Bloom BS (1956) *Taxonomy of Educational Objectives. Vol. 1: Cognitive Domain*. McKay
- Bondi AB, Ros JP (2009) Experience with training a remotely located performance test team in a quasi-agile global environment. In: Proceedings of the 2009 Fourth IEEE International Conference on Global Software Engineering - ICGSE'09. IEEE Computer Society, Washington, DC, USA. pp 254–261
- Britto R, Mendes E, Börstler J (2015) An empirical investigation on effort estimation in agile global software development. In: Proceedings of the IEEE International Conference on Global Software Engineering - ICGSE'15
- Britto R, Freitas V, Mendes E, Usman M (2014) Effort estimation in global software development: A systematic literature review. In: Global Software Engineering (ICGSE), 2014 IEEE 9th International Conference On. pp 135–144
- Chiswick BR, Miller PW (2004) Linguistic Distance: A Quantitative Measure of the Distance Between English and Other Languages. IZA Discussion Papers 1246, Institute for the Study of Labor (IZA)
- Conchúir E, Ågerfalk PJ, Holmström H, Fitzgerald B (2009) Global software development: Where are the benefits? *Commun ACM* 52(8):127–131
- Da Silva FQB, Costa C, França ACC, Prikladinicki R (2010) Challenges and solutions in Distributed Software Development Project Management: A systematic literature review. In: Proceedings of 5th International Conference on Global Software Engineering - ICGSE 10, Princeton, USA. pp 87–96
- DeLuca D, Valacich JS (2005) Outcomes from conduct of virtual teams at two sites: Support for media synchronicity theory. In: Proceedings of the 38th Annual Hawaii International Conference on System Sciences - HICSS'05. IEEE Computer Society, Washington, DC, USA. pp 50–60
- Dennis AR, Fuller RM, Valacich JS (2008) Media, tasks, and communication processes: A theory of media synchronicity. *MIS Q* 32(3):575–600
- Dictionaries O (2010) *Oxford Dictionary of English*. OUP Oxford
- Ebert C, De Neve P (2001) Surviving global software development. *IEEE Softw* 18(2):62–69
- Espinosa JA, Nan N, Carmel E (2007) Do gradations of time zone separation make a difference in performance? a first laboratory study. In: Second IEEE International Conference on Global Software Engineering - ICGSE'07. pp 12–22
- Fernandez DJ, Fernandez JD (2008) Agile Project Management - Agilism Versus Traditional Approaches. *J Comput Inf Syst* 49(2):10–17
- Gandomani T, Zulzalil H (2013) Obstacles in moving to agile software development methods; at a glance. *J Comput Sci* 9(5):620–625
- Gumm DC (2006) Distribution Dimensions in Software Development Projects: A Taxonomy. *IEEE Softw* 23(5):45–51
- Herbsleb J, Moitra D (2001) Global software development. *IEEE Softw* 18(2):16–20
- Herbsleb JD, Mockus A (2003) An empirical study of speed and communication in globally distributed software development. *IEEE Trans Softw Eng* 29(6):481–494
- Herbsleb JD, Mockus A, Finholt TA, Grinter RE (2000) Distance, dependencies, and delay in a global collaboration. In: Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work - CSCW'00. ACM, New York, NY, USA. pp 319–328
- Hofner G, Mani VS, Nambiar R, Apte M (2011) Fostering a High-Performance Culture in Offshore Software Engineering Teams Using Balanced Scorecards and Project Scorecards. In: Proceedings of IEEE 6th International Conference on Global Software Engineering - ICGSE'11. IEEE, Helsinki, Finland. pp 35–39
- Hofstede G, Hofstede GJ, Minkov M (2010) *Cultures and Organizations: Software of the Mind*. 3rd edn. McGraw-Hill
- Hossain E, Ali Babar M, Paik HY (2009) Using scrum in global software development: A systematic literature review. In: Proceedings of 4th IEEE International Conference on Global Software Engineering - ICGSE '09, Limerick, Ireland. pp 175–184
- Institute PM (2013) *A Guide to the Project Management Body of Knowledge (PMBOK® Guide)*. PMI Standard. Project Management Institute, Incorporated

- Jaanu T, Paasivaara M, Lassenius C (2012) Near-synchronicity and distance: Instant messaging as a medium for global software engineering. In: *Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference On*. pp 149–153
- Jalali S, Wohlin C (2012) Global software engineering and agile practices: a systematic review. *J Softw Evol Process* 24(6):643–659
- Kuhrmann M, Mendez Fernandez D (2015) Systematic software development: A state of the practice report from Germany. In: *Global Software Engineering (ICGSE), 2015 IEEE 10th International Conference On*. pp 51–60
- Kwasnik BH (1999) The role of classification in knowledge representation and discovery. *Library Trends* 48(1):22–47
- Laurent P, Mader P, Cleland-Huang J, Steele A (2010) A Taxonomy and Visual Notation for Modeling Globally Distributed Requirements Engineering Projects. In: *Proceedings of 5th IEEE International Conference on Global Software Engineering - ICGSE'10, Princeton, USA*. pp 35–44
- Linnaeus C (1758) *System of Nature Through the Three Kingdoms of Nature, According to Classes, Orders, Genera and Species, with Characters, Differences, Synonyms, Places (in Latin)*. 10th edn. Laurentius Salvius
- Lutz B (2009) Linguistic Challenges in Global Software Development: Lessons Learned in an International SW Development Division. In: *Proceedings of 4th IEEE International Conference on Global Software Engineering - ICGSE'09, Limerick, Ireland*. pp 249–253
- Marques AB, Carvalho JR, Rodrigues R, Conte T, Prikladnicki R, Marczak S (2013) An Ontology for Task Allocation to Teams in Distributed Software Development. In: *Proceedings of 8th International Conference on Global Software Engineering - ICGSE'13, Bari, Italy*. pp 21–30
- Mishra A, Mishra D (2014) Cultural issues in distributed software development: A review Vol. 8842. pp 448–456
- Moffitt TE (1993) Adolescence-limited and life-course-persistent antisocial behavior: A developmental taxonomy. *Psychol Rev* 100(4):674–701
- Narasipuram M (2006) Towards a Taxonomy for Globally Distributed Work. In: *Proceedings of the Twelfth Americas Conference on Information Systems - AMCIS'06, Acapulco, Mexico*. pp 867–871
- Pfleeger SL (1999) Albert Einstein and empirical software engineering. *Computer* 32(10):32–38
- Ramasubbu N, Balan RK (2007) Globally distributed software development project performance: An empirical analysis. In: *Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering - ESEC-FSE'07*. pp 125–134
- Ramasubbu N, Balan RK (2012) Overcoming the challenges in cost estimation for distributed software projects. In: *Proceedings of 34th International Conference on Software Engineering - ICSE'12, Zurich, Switzerland*. pp 91–101
- Ramasubbu N, Bharadwaj A, Tayi GK (2015) Software process diversity: conceptualization, measurement, and analysis of impact on project performance. *MIS Quarterly* 39(4):787–807
- Ramasubbu N, Cataldo M, Balan RK, Herbsleb JD (2011) Configuring global software teams: A multi-company analysis of project productivity, quality, and profits. In: *Proceedings of the 33rd International Conference on Software Engineering - ICSE'11*. pp 261–270
- Robinson M, Kalakota R (2010) *Offshore Outsourcing: Business Models, ROI and Best Practices*. 2nd ed. Mivar Press
- Royce WW (1987) Managing the development of large software systems: Concepts and techniques. In: *Proceedings of the 9th International Conference on Software Engineering. ICSE'87. IEEE Computer Society Press, Los Alamitos, CA, USA*. pp 328–338
- Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int J Comput Vis* 47(1-3):7–42
- Šmite D (2006) Global software development projects in one of the biggest companies in Latvia: Is geographical distribution a problem? *Softw Process Improvement Pract* 11(1):61–76
- Šmite D, Wohlin C, Galvina Z, Prikladnicki R (2014) An empirically based terminology and taxonomy for global software engineering. *Empir Softw Eng* 19(1):105–153
- Sommerville I (2010) *Software Engineering*. 9th edn. Addison-Wesley
- Tautz C, Von Wangenheim CG (1998) REFSENO: a representation formalism for software engineering ontologies. Technical report, Fraunhofer IESE
- Vegas S, Juristo N, Basili VR (2009) Maturing software engineering knowledge through classifications: A case study on unit testing techniques. *Softw Eng IEEE Trans* 35(4):551–565
- Vessey I, Ramesh V, Glass RL (2005) A unified classification system for research in the computing disciplines. *Inf Softw Technol* 47(4):245–255
- Vijayarathy L, Butler C (2015) Choice of software development methodologies - do project, team and organizational characteristics matter? - preprints. *Softw IEEE* 1(1)
- Vizcaíno A, García F, Caballero I, Villar JC, Piattini M (2012) Towards an ontology for global software development. *IET Softw* 6(3):214
- Wheaton GR (1968) Development of a taxonomy of human performance: A review of classificatory systems relating to tasks and performance. Technical report. American Institute for Research, Washington DC
- Wohlin C (2014) Writing for synthesis of evidence in empirical software engineering. In: *Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement - ESEM '14. ACM, New York, NY, USA*. pp 46–1464