
# Revisiting the Global Software Engineering Terminology

**Paolo Tell**
**Rosalba Giuffrida**
**Hina Shah**

**IT University Technical Report Series** **TR-2013-168**

**ISSN 1600–6100** **December 2013**

**ISSN 1600–6100**

**ISBN 978-87-7949-306-3**

**Copies may be obtained by contacting:**

**IT University of Copenhagen**
**Rued Langgaards Vej 7**
**DK-2300 Copenhagen S**
**Denmark**

**Telephone:** **+45 72 18 50 00**
**Telefax:** **+45 72 18 50 01**
**Web** **`www.itu.dk`**

# Revisiting the Global Software Engineering Terminology

**Paolo Tell · Rosalba Giuffrida · Hina Shah**

**Abstract** Even though Global Software Engineering (GSE) has been a research topic of interest for many years, some of its ground terminology is still lacking a unified, coherent, and shared definition and/or classification. The purpose of this report is to collect, outline, and relate several fundamental and discussed terms to form an initial body of knowledge. In particular, we define the diverse flavor of the GSE term relating its variations to the business, project, and team perspectives. Furthermore, we provide an overview of the various uses of the terms cooperation, collaboration, coordination, communication, and awareness taken from the Computer-Supported Cooperative Work (CSCW) tradition. This report represents on the one hand a starting point to help stimulate fruitful discussions in the relevant communities with regards to the more controversial and delicate topics, while on the other hand a coherent reference for those readers seeking clarity.

**Contents**



P. Tell
Software System Section
IT University of Copenhagen, Denmark
E-mail: pate@itu.dk

R. Giuffrida
Software System Section
IT University of Copenhagen, Denmark
E-mail: rogi@itu.dk

H. Shah
College of Computing
Georgia Institute of Technology, Atlanta, GA, U.S.A.
E-mail: hinashah@cc.gatech.edu

## 1 Introduction

Global Software Engineering (GSE) has emerged strongly as a research field. Interdisciplinary in its nature, it collects interest from diverse communities. Starting from being the focus of a workshop at the International Conference of Software Engineering (ICSE), GSE grew quickly to a point that an independent conference was established in 2006 to bring together researchers and practitioners interested in the research area.

Differently from other sciences such as medicine but similarly to other software engineering areas, the GSE research appears to be less concerned about adopting a standardized terminology. Having a unified, coherent, and shared set of terms to describe the different concepts and terms allows to avoid possible misunderstandings and permits outsiders to quickly grasp the vocabulary necessary to discourse in the field. Moreover, a standardized set of terms allows for an efficient indexing of the knowledge body, which eventually improves the chances of easily performing accurate searches through digital libraries.

Motivated by these perceived difficulties of finding clear definitions and classifications, which often result in the realization of a concrete lack of material, here we attempt to improve the GSE body of knowledge. The knowledge herein reported will either duly reference and report from the literature—whenever possible—or provide a dialectic argument supporting the definitions and relations presented.

This report has been designed mainly to tackle two objectives:

- on the one hand, we tackled a gap that we think is necessary to fill, hence, we provide a solid and comprehensive reference for the terminology used in GSE. In particular, we define the diverse flavor of the GSE term relating its variations to the business, project, and team perspectives; and, we provide an overview of the various uses of the terms cooperation, collaboration, coordination, communication, and awareness taken from the Computer-Supported Cooperative Work (CSCW) tradition.
- on the other hand, we published this piece of work as a technical report in the hope that it can represent a starting point for stimulating fruitful discussions in the relevant communities with regards to the more controversial and delicate topics.

## 2 What is GSE?

In the last decades, the development of software has rapidly moved from being solely an activity performed by one single individual, to an activity involving a group of individuals, a team, working together to pursue a purpose (i.e., the creation of a piece of software). This phenomenon, which nowadays is a well established approach in software engineering, is known as either Global Software Engineering (GSE) or Global Software Development (GSD) depending on whether the focus is on the application of engineering principles to the software creation or on the software creation itself. This differentiation might find disagreements among readers; however, if a distinction is deemed necessary, it is important to realize that software engineering activities include software development ones and not vice versa.

| | |
|---|---|
| **Sourcing** | |
| Insourcing | The process of acquiring a service to be performed in-house. This contracting form can also be used internally across different company divisions/teams as long as there is a contractual arrangement. |
| Outsourcing | The process of delegating a specific task to an external company. |
| **Shoring** | |
| Inshoring | (or onshoring in [30]) When the client of the contract is located in the same country. |
| Offshoring | When the client on the contract is located in a different country. |
| Nearshoring | When the two parties of the contract are located in different countries (offshoring), which are geographically close. |
| Farshoring | When the two parties of the contract are located in different countries (offshoring), which are geographically distant. |

Table 1: The business perspective: sourcing and shoring arrangements (inspired from [30]).

The GSE phenomenon has had from the beginning a tremendous impact on companies. Terms like outsourcing quickly became common terms also in the software engineering fields; and companies rapidly needed to adapt to face an evolving market. This shift was followed quickly by a strong demand of the industry to academia with regards to the need of finding best practices (e.g., [15]), tools for assessing risks on engaging in more competitive contracting endeavors (e.g., [18]), lessons learned (e.g., [24]), and so on. Several terms are used and represent this **business perspective** (Table 1). All these terms have a longer history in business, hence, should not need a clarification; however, they are worth mentioning as they are still used and sometimes misused in GSE related research. As per our knowledge, [30] is the first work showing these definitions in the GSE field, which was further on extended in [31]. Differently from [30] and [31], however, we would like to emphasize that these terms are meant to describe business arrangements rather than forms of collaboration. Nonetheless, the structure is kept from [30]. Therefore, the terms are divided in two groups, namely sourcing and shoring [30]. The former refers to the form of contracting the business process; the latter reflects the distance or location of the client (in the contract).

Looking at the object of the business arrangement, we now move from the business perspective to the **project perspective**: the one where the focus is the software engineering project. With this focus, two characteristics can be used to describe the software engineering project allowing the provision of a high level classification of software engineering arrangements often used to discourse on GSE: the number of people involved in a given software engineering project and the number of physical locations comprising such endeavor. Figure 1 shows the matrix resulting by considering such characteristics. Disregarding the combination of a single person working in multiple sites, which is unimportant in this discourse, the matrix shows that a software engineering endeavor from being an individual project becomes a co-located team project when it involves multiple people cooperating, and the co-located team project becomes distributed when such people or teams are distributed rather than co-located. Even if intuitively correct, this classification finds conflicting opinions regarding the discriminating property that allows to distinguish a co-located arrangement from a distributed one as well as the terminology that identifies the different spatial distributions of a team. In the next two paragraphs we will try to bring together the literature from multiple

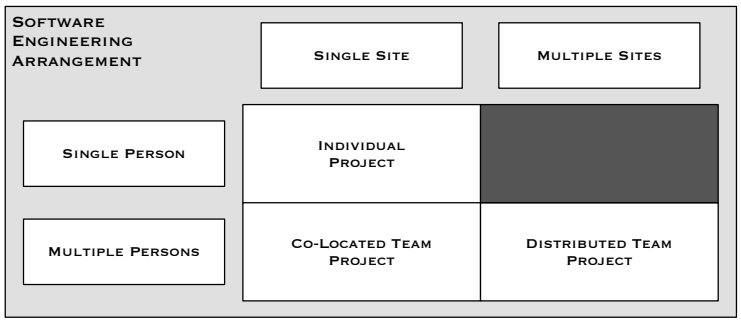


Fig. 1: The project perspective: a classification of software engineering arrangements.

domains to either describe the research done in such area or attempt to clarify ambiguity.

Analyzing teams composed by people located more than 30 meters apart, Allen [1] demonstrated that the communication frequency of such teams was closer to the one of teams composed by people located kilometers away rather than the ones in which people are co-located. Results that have been corroborated by a later study by Kraut et al. [16], which was focused on collaboration among scientists. Without entering the debate of whether a distance that could be perceived as insignificant as 30 meters is the threshold that represents the shift from co-located to distributed arrangements, it is important to recognize that these two team configurations presents unique characteristics, hence, challenges. Quoting Herbsleb [14], "many of the mechanisms that function to coordinate the work in a co-located setting are absent or disrupted in a distributed project".

Moreover, different arrangements can be identified when focusing on the structure of the teams. To the best of our knowledge, a terminology to distinguish the various spatial distribution of teams is missing in the literature. For instance, researchers have been using indiscriminately the terms geographically distributed and geographically dispersed. Three kinds of distribution are identified by Giuffrida et al. [13] and reported in Table 2, which we call the **team perspective**. For each of these team structures, examples from the GSE field are provided.

Each of the software engineering arrangements pictured in Figure 1 is affected by a specific set of challenges, which can jeopardize the successful fulfillment of those endeavors. Focusing on the dimensions of distribution, we can further analyze different arrangements of distributed team projects with peculiar characteristics. According to Noll et al. [22], the distances that provide barriers to collaboration[1] in GSE are geographical, temporal, cultural, and linguistic. Using such classification, we have further detailed the last quadrant of Figure 1. Figure 2 shows how different distributed team projects are affected by the four distances. In particular, in generic distributed software engineering arrangements, any of such distances can

[1] A reader might be wondering about the indistinct use of the collaboration and cooperation terms. These concepts are purposefully left undefined, a clarification about the terminology regarding these two concepts as well as the ones of coordination, communication, and awareness will be covered in the next section. Until then, we ask the reader to accept the fact that when referring to the work of other researchers the term used will have to respect the original work, thus potentially differ from cooperation.

| | |
|---|---|
| Distributed | Teams are distributed around the globe. The size and number of teams can vary. Co-located physical collaboration happens in each site and global virtual collaboration happens across sites. This is the most widely investigated arrangement and several studies of this setup have been published (e.g.,[12], [4], [27], [21]). |
| Dispersed | Two or more team members are geographically distributed and team members tend to be physically alone — thus dispersed. This is a common practice in Open Source (OSS) or for example when the access to most talented developers is needed. From the best of our knowledge, teams of this kind have been mainly investigated in OSS (e.g., [28]). |
| Partially-dispersed | One co-located team collaborates with one or more globally dispersed team members. There is at least one co-located team, but there can be multiple distributed teams. Co-located physical collaboration happens in the team, while global virtual collaboration happens among dispersed team members and between the core team and the disperse members (e.g., [29], [8]). |

Table 2: The team perspective: distributed, dispersed, and partially-dispersed [13].

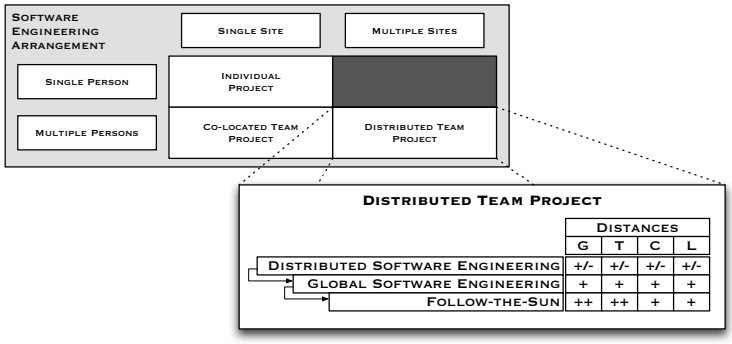


Fig. 2: Distributed software engineering arrangements.

be either present or not; to some extent, all of them need to be present in GSE arrangements; whereas, a preponderant influence of geographical and temporal distances affect the special strategy of GSE arrangements known as follow-the-sun. Not surprisingly, this set of distances is not accepted by all researchers. As an example, in [5] the linguistic distance is seen only as one of the forms of cultural distance, which for the authors includes also "[...] ethnic differences, national and political differences, individual perceptions and motivation, and work ethics". Despite the classification of distances chosen, it is important to realize that each of the different arrangements (and often each specific case) is affected by specific challenges and requires appropriate support.

Differently from individual ones, projects involving multiple persons are hindered by factors that, rather than being related to mere software engineering concerns, are idiosyncratic challenges of human interactions. Not surprisingly, an extensive amount of research focuses on supporting cooperation in software engineering. Based on the arrangements depicted in Figure 1, Figure 3 illustrates the tool support required by each arrangement and emphasizes that each each of them incrementally builds on the needs of the previous one as indicated by the arrow. In particular, an individual project mainly requires tools specific to soft-

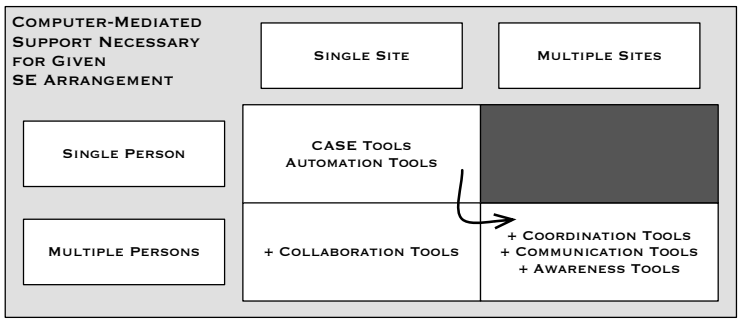


Fig. 3: Needs related to the different software engineering arrangements.

ware engineering (CASE[2] tools) and tools able to automate processes like build, testing, etc. Continuing along the arrow, in addition to the requirements of an individual project, a team project greatly benefits from tools able to provide a shared workspace, a virtual space containing all the artifacts related to a project accessible to all team members. These softwares are known as collaboration tools, i.e., a version control system. Finally, at the tip of the arrow, distributed software project. These arrangements are characterized by additional challenges related to the ability of coordinating, communicating, and being aware[1]. Examples of computer-mediated means that support these shortages range from calendaring systems (coordination tools), messaging or video-conferencing systems (communication tools), to systems able to improve the individual perception of information (awareness tools).

In the next section we will analyze the concepts of cooperation, communication, collaboration, coordination, and awareness, which have been mentioned in this section. These are extensively used terms in the GSE community; however, maybe because the GSE community is younger than the communities from which these concepts are borrowed, their use is often imprecise.

## 3 CSCW Concepts for GSE

In 1991, Ellis et al. [9] proposed that "to support group interaction, we must attend to three areas: communication, collaboration and coordination". Later, these three dimensions were named *the 3C collaboration model* by Fuks et al. [10], which further elaborated and extended the model. In particular, three modifications were applied. Firstly, the term collaboration was defined as the union of communication, cooperation, and coordination efforts, hence, reshaping the original interpretation presented in [9]; secondly, the whole model was named the collaboration model, and the collaboration dimension was substituted by the cooperation one, addressing the "joint operation in a shared workspace"; thirdly, the awareness dimension was incorporated as part of the 3C model by relating it to the other 3Cs.

Often in GSE, researchers indiscriminately refer to the 3C collaboration model as either [9] or [10] without explicitly addressing the differences between the two

[2] CASE tools: computer-aided software engineering tools

| | |
|---|---|
| Cooperative work | General and neutral designation of multiple persons working together to produce a product or service. |
| Collaborative work | A special stress is given to a particular "collaborative" or complying spirit among the cooperators, as evident, for example, in the expression "collaborating" with an enemy. |
| Collective work | Cooperative work where the cooperating ensemble is sharing the responsibility for accomplishing the task. The emphasis of the concept is on the fusion of the members of the ensemble into a whole, a "collective". That is, the term is conceptually close to "group" and "team" work. |

Table 3: Definitions of forms of Cooperative Work [2].

models. For instance, independently from the source—either [9] or [10]—the 3C model is widely used in the GSE literature to analyze both tools (e.g. [32]) and practices (e.g. [19], [20]). Further complicating the scenario, another "C" was introduced by Carmel [3]—the control dimension, which is defined as the "process of adhering to goals, policies and standards, or quality level" [3], . It is important to emphasize that neither [9] nor [10] are referred by Carmel as well as no relationship between his terms and the 3C is made in [3]. According to Carmel, the distances in GSE affects communication, coordination, and control. On the other hand, these dimensions are often considered as three different forms of cooperation within a team (e.g., [17]). Summarizing, when researchers use the 3C model in GSE, they can refer to Ellis, Fuks, or Carmel dimensions. Therefore, if on the one hand, communication and coordination are certainly two of the 3C, the third C can be, depending on the model adopted, collaboration, cooperation, or control. While control refers to a management perspective that is out of the focus of this report and will not be further considered in the following, the distinction between collaboration and cooperation deserves further discussion.

Cooperation and collaboration are terms often used in an interchangeable way in GSE. However, the CSCW community has a long tradition dealing with terms such as cooperative work and collaborative work as well as collective work and group work. In particular, Bannon and Schmidt unpacked these concepts in the CSCW field already in 1989 [2]; according to their work (Table 3), there are many forms of cooperative work and several terms can be used depending on the perspective that needs to be highlighted (i.e., collaborative, collective, and group work). According to a later work of Schmidt [25], "people engage in cooperative work when they are mutually dependent in their work and therefore are required to cooperate in order to get the work done. [...] Being mutually dependent in work means that A relies positively on the quality and timeliness of B's work and vice versa. B may be 'down stream' in relation to A but in that case A nonetheless will depend on B for feedback on requirements, possibilities, quality problems, schedules etc.". Following the definition by Schmidt [25] and supporting the claim by Dittrich et al. [6], we consider software engineering work cooperative in nature. We argue that collaboration, coordination, communication, and awareness are four dimensions of cooperative work[3].

On the one hand, from a tool perspective, in cooperative work, for the purpose of collaborating on a common product, two (or more) persons involved in an ac-

[3] Please note that the three dimensions proposed by Ellis [9] are integrated with the awareness dimension introduced by Fuks [10].

| | |
|---|---|
| Collaboration | When the technology brings improvements to the shared space or to the way users interact with shared artifacts synchronously or asynchronously. |
| Coordination | When the technology brings improvements to the support offered for people managing themselves, or themselves within a team. |
| Communication | When the technology brings improvements to the way messages and information are exchanged among people, reducing gaps, ambiguity, or the effort needed to understand, establish, or continue a conversation. |
| Awareness | An understanding of the activities of others, which provides a context for your own activity. |

Table 4: Operational definitions from a tool perspective (inspired or taken from [7][9][32]).

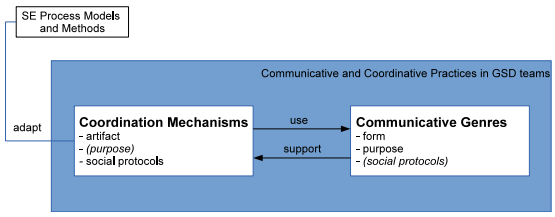


Fig. 4: Theoretical framework proposed for analyzing cooperative practices in GSD [11].

tivity need to coordinate their actions. This coordination process entails engaging in direct communication in the case where one of the persons involved in the coordination is not sufficiently aware of what is required to work on the product (e.g., due to doubts, misunderstandings, and misalignments). The cooperation scenario suggested above links together in a coherent manner the four terms; however, an operational definition for each of them is required to classify the numerous tools that have been developed to support cooperation in GSE. The 3C model has often been used to partially fulfill such an objective. Starting from such knowledge [7][9][32], we provide the four operational definitions in Table 4.

On the other hand, from a practice perspective, collaboration, communication, coordination and awareness have all different connotations and designate different types or facets of cooperative work [2]. These elements are visible in the actual practices and the terms can be used to refer to certain aspects of cooperative work; however, the 3C collaboration model is not sufficient to describe and analyze cooperative practices in GSE. Thus, more complex theoretical concepts developed in other communities (e.g., CSCW, IS, CHI) need to be taken into consideration to look at cooperative practices, going beyond the 3C collaboration model as it has been adopted to date. For example, Giuffrida and Dittrich [11] proposed to analyze cooperative practices in GSE looking at coordinative and communicative practices, and using the analytical tools of coordination mechanisms [26] and communicative genres [23], as illustrated in Figure 4. Explicit communication through communication channels provides explicit awareness; when established coordination mechanisms support the cooperation, communication, and awareness are effectively mediated by artifacts. Despite the terminology used is similar to the one suggested by the 3C collaboration model, the specific concepts function as an analytical tool to study cooperative practice.

**4 Acknowledgements**

The authors wish to thank Dr. Yvonne Dittrich and Dr. Muhammad Ali Babar for their feedback and discussion on the topic.

**References**

1. T. J. Allen. Managing the flow of technology: technology transfer and the dissemination of technical information within the R&D organization. *Cambridge (MA)*, 1977.
2. L. J. Bannon and K. Schmidt. Cscw-four characters in search of a context. *DAIMI Report Series*, 1989.
3. E. Carmel and R. Agarwal. Tactical approaches for alleviating distance in global software development. *Software, IEEE*, 2001.
4. D. Damian, L. Izquierdo, J. Singer, and I. Kwan. Awareness in the wild: Why communication breakdowns occur. In *Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pages 81–90. IEEE, 2007.
5. S. Deshpande, I. Richardson, V. Casey, and S. Beecham. Culture in Global Software Development - A Weakness or Strength? In *International Conference on Global Software Engineering*, 2010.
6. Y. Dittrich, D. Randall, and J. Singer. Software Engineering as Cooperative Work. *Computer Supported Cooperative Work (CSCW)*, 2009.
7. P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*. ACM, 1992.
8. M. S. Elliott and W. Scacchi. Free software developers as an occupational community: resolving conflicts and fostering collaboration. In *Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work*, pages 21–30. ACM, 2003.
9. C. A. Ellis, S. J. Gibbs, and G. Rein. Groupware: some issues and experiences. *Communications of the ACM*, 1991.
10. H. Fuks, A. B. Raposo, M. A. Gerosa, and C. J. Lucena. Applying the 3c model to groupware development. *International Journal of Cooperative Information Systems*, 2005.
11. R. Giuffrida and Y. Dittrich. A conceptual framework to study the role of communication through social software for coordination in globally-distributed software teams. *Submitted to nformation and Software Technology Journal, March 2014*.
12. R. Giuffrida and Y. Dittrich. Exploring the role of social software in global software development projects. In *Global Software Engineering Workshop (ICGSEW), 2011 Sixth IEEE International Conference on*, pages 108–110. IEEE, 2011.
13. R. Giuffrida, Y. Dittrich, and H. Sharp. You can not ask what you do not suspect - an argument for observational studies in global software development research. *Submitted to Empirical Software Engineering Journal, March 2014*.
14. J. Herbsleb. Global software engineering: The future of socio-technical coordination. *2007 Future of Software Engineering*, 2007.
15. K. Heston and W. Phifer. The multiple quality models paradox: how much 'best practice' is just enough? *Journal of Software Maintenance and Evolution: Research and Practice*, 2010.
16. R. Kraut, C. Egido, and J. Galegher. Patterns of contact and communication in scientific research collaboration. In *Proceedings of the 1988 ACM conference on Computer-supported cooperative work*, 1988.
17. F. Lanubile. Collaboration in distributed software development. In *Software Engineering*. 2009.
18. C. Magnusson and S.-C. Chou. Risk and Compliance Management Framework for Outsourced Global Software Development. In *International Conference on Global Software Engineering, 2010*.
19. C. Manteli, H. v. Vliet, and B. v. d. Hooff. Adopting a social network perspective in global software development. In *International Conference on Global Software Engineering, 2012*.
20. S. Modi, P. Abbott, and S. Counsell. Negotiating common ground in distributed agile development: A case study perspective. In *International Conference on Global Software Engineering, 2013*.

21. N. B. Moe, G. K. Hanssen, et al. From offshore outsourcing to offshore insourcing: Three stories. In *Global Software Engineering (ICGSE), 2012 IEEE Seventh International Conference on*, pages 1–10. IEEE, 2012.
22. J. Noll, S. Beecham, and I. Richardson. Global software development and collaboration: barriers and solutions. *ACM Inroads*, 2010.
23. W. J. Orlikowski and J. Yates. Genre repertoire: The structuring of communicative practices in organizations. *Administrative science quarterly*, 1994.
24. L. Pilatti and J. L. N. Audy. Global Software Development Offshore Insourcing Organizations Characteristics: Lessons Learned from a Case Study. In *International Conference on Global Software Engineering, 2006*.
25. K. Schmidt. Cooperative work and its articulation: requirements for computer support. *Le Travail Humain*, 1994.
26. K. Schmidt and C. Simone. Coordination mechanisms: Towards a conceptual foundation of cscw systems design. *Computer Supported Cooperative Work (CSCW)*, 1996.
27. H. Shah, S. Sinha, and M. J. Harrold. Outsourced, offshored software-testing practice: Vendor-side experiences. In *Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on*, pages 131–140. IEEE, 2011.
28. H. Sharp, R. Giuffrida, and G. Melnik. Information flow within a dispersed agile team: A distributed cognition perspective. *Agile Processes in Software Engineering and Extreme Programming*, pages 62–76, 2012.
29. A. Sigfridsson and A. Sheehan. On qualitative methodologies and dispersed communities: Reflections on the process of investigating an open source community. *Information and Software Technology*, 53(9):981–993, 2011.
30. D. Smite and C. Wohlin. A Whisper of Evidence in Global Software Engineering. *Software, IEEE*, 2011.
31. D. Šmite, C. Wohlin, Z. Galviņa, and R. Prikladnicki. An empirically based terminology and taxonomy for global software engineering. *Empirical Softw. Engg.*, 2012.
32. I. Steinmacher, A. Chaves, and M. A. Gerosa. Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature. *Computer Supported Cooperative Work (CSCW)*, 2013.