Distribution Dimensions in Software Development Projects: A Taxonomy

Dorina C. Gumm, University of Hamburg

For many economic and technological reasons, companies are increasingly conducting projects on a global level. Global projects are highly distributed, with experts from different companies, countries, and continents working together. Such distribution requires new techniques for project coordination, document management, and communication. Distribution complexities include various project types—such as global, interorganizational, or open source software projects—that are distributed in different ways and face particular challenges. Some researchers focus on problems of global distribution and related lessons learned, whereas others emphasize distribution as an advantage in software development that speeds the development process—for example, in OSS projects.

However diverse the project and whatever the perceived benefits or drawbacks, understanding the underlying distribution phenomenon is crucial to analyzing existing methods and tools for their applicability in distributed project settings. This understanding is also important when proposing new methods or tools. To negotiate distributed project requirements, for example, requires an understanding of distributed project stakeholders. Likewise, requirement document consistency might be related to distributed artifacts.

In this article, I seek to explain the distributed software development phenomenon. My goal is to provide developers and other stakeholders a foundation for discussing specific challenges they might face in distributed software development projects. To this end, I present a taxonomy based on an earlier literature study. Here, I continue this earlier work by using a case study to exemplify the taxonomy’s use and substantiate its value.

Distributed software development literature

Understanding distributed project issues is a challenge not just for global software development, but also for software research areas such as OSS development and computer-supported cooperative work. In addition, several papers in the literature consider distribution from specialized software engineering fields, such as requirements engineering and part-
Regardless of the differing viewpoints, in all areas technical solutions as well as social challenges are discussed.

Most existing work reports on case studies that either faced special challenges of distributed people or artifacts or found solutions for specific problems. Some researchers aim at a broader view: J. Roberto Evaristo and Richard Scudder identify distributed project dimensions, while Rafael Prikladnicki and his colleagues refine that work and provide a physical distribution typology. In other work, Maria Paasivaara identifies types of distributed projects.

In my literature analysis, I found that researchers discuss distribution topics from several different perspectives, each of which entails specific issues:

- Who or what is distributed? For example, when people are distributed, challenges include coordinating communications and workflow, and managing cultural differences. Distributed artifact challenges relate to who's responsible for various artifacts and their version control.

- In what way are people and other entities distributed? For example, in some projects, it's important to distinguish between stakeholders distributed among different geographical locations and those distributed among different companies.

- What are the distribution's specific challenges? Challenges might include the previously mentioned version control issues, cultural differences, or coordination of synchronous or asynchronous communication.

- How can distribution challenges best be solved? Developers address some challenges through social solutions (such as project management and mutual understanding) while others require technical solutions (such as tools for coordination and communication and for managing the programming process). This technology use can in turn create new distribution problems as well as create and support distributed project settings.

Figure 1 shows the connections between these perspectives: distributed entities cause specific challenges which in turn are handled by specific solutions. Such solutions might then give rise to new challenges. For example, some solutions, such as Internet-based communication tools, can both create and support distributed project settings.

**Dimensions of distribution: A taxonomy**

To clarify the distribution phenomenon—and thus create a preliminary taxonomy—I distinguish issues that describe distribution itself from issues that distribution creates in the project setting. The dimensional description refers to distribution itself and so to the questions of who or what is distributed and in what way. I don't include individual challenges and solutions because they result from distribution rather than being part of the phenomenon itself.

The taxonomy uses *dimensions* to describe the ways in which people or artifacts are distributed. According to the Merriam-Webster Online Dictionary, *dimension* typically refers to a physical property and is "the range over which or the degree to which something extends." The term *distribution* describes "the position ... of occurrence (as of the members of a group) over an area or throughout a space or unit of time." Because distribution occurs in different degrees and, like space, seems to occur along a continuum, it seems adequate to describe distribution using the dimension concept.

In software development projects, there are four basic distribution dimensions, each of which can range from lightly to heavily distributed. According to the literature, those dimensions are

- physical (or geographical) distribution,
- organizational distribution,
- temporal distribution, and
- distribution among stakeholder groups.

Individuals and stakeholder groups can be distributed among physical space, organizations,
and time, whereas other entities can be distributed among physical space, organizations, and stakeholder groups. Figures 2 and 3 from the case study (p. 49) illustrate these findings.

**Physical distribution**

Physical distribution is one characteristic of distributed projects. People and things can be distributed among different physical locations and to different degrees. In some projects, distributed people or objects among different floors in the same building can be a challenging distance. Other projects deal with location distributions in the same city, among different cities or countries, or even among different continents.

Physical distribution of people is the literature’s most discussed topic. Such distribution occurs both in global software development, where organizations from different countries work together as partners or suppliers, as well as in large OSS projects with numerous international contributors. Prikladnicki and colleagues have developed a detailed typology for physical distribution among users, customers, and project teams, as well as within project teams.

**Organizational distribution**

Organizational distribution is another characteristic of distributed projects. Organizational distribution is related to the structures people work in on the project, but these structures don’t necessarily represent their everyday work or employer environments. Although the structure might refer to a company or noncommercial organization, it might also refer to any other project structure that describes the organized work’s condition or state.

The target organizations might be partners, subcontractors, suppliers, customers, or consultants. In customer-specific software development projects, the typical organizations involved are the software supplier, the customer’s company, and a consulting firm. Developers in OSS development projects often simultaneously work for different companies. However, these companies don’t typically play a specific project role, but rather represent the developers’ background experience. Another example of organizational distribution is when several divisions of one company are involved in a project. Finally, many research projects involve people from different institutions working together on a particular problem.

**Temporal distribution**

Researchers have also identified temporal distribution as a primary characteristic of distributed projects. This type of distribution refers to work-hour synchronicity—that is, the time during which project members are available for real-time interactions. Time separations might be rooted in physical distribution across different time zones. It might also result from shift work or varying working rhythms, as when people work part-time on OSS projects. In the latter case, time separation is viewed as an advantage because it might speed development.

**Distribution across stakeholder groups**

Artifacts, skills, and other entities can be distributed not only across locations and organizations, but also among stakeholder groups. Requirements specifications, for example, are often distributed among users, managers, analysts, and developers. Such distribution requires significant document management effort when requirements are documented from different viewpoints, at different detail levels, and with varying documentation tools. The literature discusses both artifact distribution and task distribution.

**Understanding distribution in a given project**

To analyze a project’s distribution, you must first identify its distributed entities along the taxonomy’s dimensions. You should base this analysis on observable project settings, such as organigrams or the locations of people and organizations. In addition, you must also account for the people’s perceived distances, which might differ from their actual distances.

**Examples of distributed entities**

Software projects might deal with distribution within single stakeholder groups or among several stakeholder groups, as well as distributed entities such as software code, documentation, tasks, or power. If a company uses software experts from different countries on a global project, physical distribution within the developer group is likely the most important factor. Also, in OSS projects, the developer community is often distributed both physically and organizationally. Distribution among stakeholder groups is given if developers and analysts reside in different countries,
which often entails physical, temporal, and organizational distribution issues.

In addition, the boundaries separating stakeholder groups are often blurred. In OSS projects, for example, users can join the development community even without having to actually program. Furthermore, participatory design approaches aim to decrease user and developer distribution so that designers participate in users’ worlds, users participate in design activities, or both. In a distributed project, if this happens successfully—that is, the development team consists of users and developers—distribution between two stakeholder groups changes to distribution within the developer group.

As for artifact distribution, a requirements specification document example illustrates several issues. In one project, the document might be organizationally distributed among customer and supplier organizations. In another project, it might be distributed only within a customer organization, but among different stakeholder groups (users and managers, for example). In this case, the physical distribution is entailed in the organizational distribution. Still, the physical distribution level might vary from a few folders on a hard disk to numerous computers in different countries. In a third project, the document might be dispersed across all dimensions.

**Perceived distance**

You can easily analyze observable project settings using project documents, for example, that describe who’s involved, where they’re located, and so on. However, the perceived distance in each dimension also plays a major role in a project’s dispersion.

Evaristo and Scudder measure people’s and stakeholder group’s dispersion level by the perceived distance among the members sharing a particular role as well as by the perceived distance between this role group and all the others. The perceived-distance concept is applicable not only to physical distribution but also to organizational or temporal distribution. Project members who are organizationally distributed but work closely together at the same place and during the same working hours might perceive the organizational distance as low, whereas members working in different locations across organizations would likely perceive the organizational distance as high. The perceived temporal distance might vary according to the project’s established communication mechanisms and culture. Team members who established a good asynchronous communication culture might perceive temporal distance as low, whereas people who prefer meetings or telephone calls might perceive temporal distance as high.

**The CommSy case study**

The University of Hamburg’s CommSy project offers a case study in how to apply the taxonomy, correlate dimensions, and relate specific challenges to those dimensions.

Launched in 1999, CommSy is a Web-based community system (or groupware) developed to support education and networked project work. Although originally started to support university education, CommSy is now used in different contexts, including schools, freelancer networks, and small companies. The project setting has also changed several times: what started as an extracurricular project evolved into a research project and is now an open source project with several smaller subprojects.

**Project characteristics**

The CommSy project consists of different independent projects whose members maintain CommSy and further develop it for special contexts. However, the project’s goal is to maintain one joint development, and I’m referring to this joint development here. Although not a global project, CommSy is highly distributed and thus a suitable example for applying the taxonomy.

One CommSy research topic is to analyze and improve the project’s requirements engineering process. CommSy has an increasing number of users, application contexts, and usage locations. This distribution complicates the participatory process of requirements understanding, and the various independent projects complicate internal requirements management. Thus, one requirements engineering challenge is to understand the character and key issues of distribution within this project.

When analyzing the CommSy project distribution, we can focus on individual subprojects or on the joint CommSy development efforts. As figures 2 and 3 show, distribution varies within the project.

**Physical distribution**

The physical distribution within the project team is rather small, as most members work in
the university’s Department of Informatics. However, some distribution remains because team members are employed by different department sections and thus work in different buildings and on different floors. The perception of distance therefore varies by team member and his or her particular project role. Also, some members have moved to positions outside the university and have either maintained close contact with project members or withdrawn from the project. Finally, some project members temporarily work in other countries. This distance is indeed a challenge, but hasn’t played a major role because it’s temporary. Generally, physical distance within the project team causes problems in information exchange and process transparency.

The physical distribution between the project team and the users, as well as among user groups, is on a higher level. Although the largest user group is in the Department of Informatics, the number of external users continually increases. Different organizations in Hamburg as well as in many other German cities use CommSy, and some user groups are even outside Germany. This physical distribution causes problems in the participatory design process. It might be that few users participate in the process because it’s such a large user group and most users simply aren’t interested. In any case, the distance hampers efforts to bring team members and users together.

Other physically distributed entities include software code and requirements. To cope with the distribution, the team maintains the software code on the SourceForge open source platform. Requirements and their documentation are also physically distributed because the team manages and documents requirements and decisions about the implementation on SourceForge as well as in the CommSy project room software and at face-to-face meetings. This distribution of requirements and related documentation necessitates extra effort to keep the requirements engineering process transparent and traceable.

**Organizational distribution**

Most team members not only work in different locations within the Department of Informatics, they also belong to different research projects and handle the projects’ respective user groups. In addition, some team members work in different disciplines. This organizational distribution creates challenges, such as commitment difficulties: it’s often hard for members to give commitments that others can count on. In
the past, this was particularly problematic during strategic planning. Organizational distribution also means that team members have different views on what's important in the developing process. Team members’ perception of organizational distance varies according to individuals and by project phase.

Among user groups, organizational distribution emerged as a bigger challenge than physical distribution. CommSy operates in many different contexts. Within the university, various disciplines—from computer science to pedagogy to linguistics—use the system. External organizations, such as companies and schools, also use CommSy. Given its wide application field, CommSy users have widely varying backgrounds and attitudes. Among the organizational distribution challenges we face are different (and sometimes contradicting) usability and functionality requirements that hamper the goal of a common system vision.

Requirements are organizationally distributed because they emerge in different user organizations where they are discussed with the dedicated team members. This not only challenges process transparency, but requires extra effort to consolidate and negotiate requirements.

Temporal distribution

The project team’s temporal distribution is rooted in members’ asynchronous working times, which is in turn rooted in the organizational distribution. In addition, most team members work only part-time on the project. Thus, common working hours are rare and make information exchange and transparent development difficult.

Temporal distribution plays a minor role in interaction between the development team and users. Such interaction consists mainly of single workshop and interview meetings, along with extended user support, all of which involve only the typical coordination problems. Because the development team mediates the interaction between user groups, the temporal distribution between user groups plays a similar minor role.

Distribution across stakeholder groups

Requirements are dispersed among stakeholder groups because they appear in different user groups and must be transferred to the project team. In turn, the project team must send requirements-handling decisions back to user groups. So, the distribution among stakeholder groups and the challenges entailed are tightly connected to the organizational distribution.

Decision-making power is also distributed among stakeholder groups. This distribution is rooted in the flat hierarchy and the project team’s voluntary status, as well as in the existence of different customers. Such distribution creates challenges regarding project management, commitments, and requirements negotiation.

Results

The CommSy case study illuminates several challenges related to physical and organizational distribution, and how project teams can deal with them.

Physical distance. Although physical distance is indeed a challenge, it is not a big one for two reasons. First, most team members are located close enough to enable regular meetings. Second, they have good, technology-based communication. The team copes with user group distance by providing personal support, organizing workshops, and conducting online surveys.

Organizational distance. The project’s organizational distance emerged as the major challenge, creating difficulties in maintaining a common product vision and in establishing process transparency. User group distribution challenges the team’s desire to use participative design methods because users work in very different contexts and thus have different viewpoints, needs, and requirements. As a result, the CommSy team must invest extra effort to evaluate conflicting requirements, make decisions transparent, and maintain a common system vision.

When improving the project’s requirements engineering process, the team must therefore pay particular attention to organizational distribution. That is, they must

■ consolidate requirements from different contexts to maintain a common system vision;
■ negotiate conflicting requirements; and
■ provide bidirectional feedback among project team members and user groups, and even among different user groups.

Current endeavors to meet these requirements include providing mediated feedback to
improve our transparent participatory process\(^{10}\) and explore commented case studies to support cross-contextual software development.\(^{20}\)

**The CommSy case study analysis offers two primary benefits. First, it shows how the taxonomy can help team members understand their project’s particular distribution issues and their scope. Second, it offers a real-world example in which organizational distribution is a bigger challenge than physical distribution. CommSy isn’t a global project. However, it raises issues that global software projects must contend with, and shows how they might use the taxonomy to understand the nature of distribution in their own projects.**

That said, the literature study I based the taxonomy on could be expanded, either within the target areas or by consulting broader literature, such as that on virtual teams or project management. In addition, the selected literature reports mainly on case studies about distribution problems and practices and doesn’t include social or organizational theories. This might be a weakness, because it likely creates an incomplete model. However, it might also be a strength, because it grounds the dimensions in practical challenges.

In any case, there is further work to do. I plan a more detailed discussion of the correlations between the dimensions, as well as of individual challenges.

---

**References**