



Team-external coordination in large-scale software development projects

Aivars Sablis¹ | Darja Smite¹ | Nils Moe^{1,2}

¹The Department of Software Engineering, Blekinge Institute of Technology, Blekinge Tekniska Högskola, Karlskrona, 371 79, Sweden

²Digital, SINTEF, Strindvegen 4, Trondheim, 7034, Norway

Correspondence

Aivars Sablis, The Department of Software Engineering, Blekinge Institute of Technology, Blekinge Tekniska Högskola, Karlskrona 371 79, Sweden.
Email: aivars.sablis@bth.se

Funding information

Research council of Norway, Grant/Award Number: 267704; Swedish Knowledge Foundation, Grant/Award Numbers: KK-Hög grant 2009/0249, KK-Hög grant 2016/0191

Abstract

High-quality work is said to depend on team abilities. However, teams working in large-scale projects often do not have all expertise to complete their tasks, which are also highly interdependent. Therefore, teams need to rely on coordination with other teams, experts, and supporting roles. In this paper, we explore teams' coordination needs and evaluate the impact of the satisfaction of these needs on team performance. We conducted an embedded multicase study with nine teams in two projects in two companies. We collected qualitative data through nine focus groups and 19 interviews and quantitative data using a questionnaire with 49 members from the studied teams. Our results suggest that project-, team-, and task-related characteristics impact teams' coordination needs. Even in the same project, teams may have different expertise and work coordination needs. We found that the satisfaction of these needs seems to influence teams' performance, although our results are inconclusive and yield a closer look in future research. On the basis of our findings, we recommend the companies to cultivate a networking culture and support teams external coordination with other teams and experts, paying attention to their needs, for example, driven by a lack of experience or increased work complexity.

KEYWORDS

case study, empirical, expertise coordination, large-scale software development, work coordination

1 | INTRODUCTION

In recent years, as software-intensive systems have grown in size and complexity, large-scale software development projects have become more and more common. Yet, many companies struggle to achieve the desired performance because of the complexity,^{1,2} the size of developed software systems,³ the number of people involved,⁴ and distribution of teams in multiple sites.^{5,6}

Coordination is one of the biggest challenges associated with large-scale software projects. Large-scale projects have large complex codebases, many teams working on the same codebase at the same time, and the need for many experts to coordinate tasks that have complex dependencies between them. In some projects, there is a need to coordinate more than a hundred developers in multiple teams, with all the external stakeholders supporting or influencing the project.⁷ When teams solve tasks in large and complex systems, they require access to and need to coordinate with many experts making decisions related to the project, for example, architecture, system design and specification, and system integration. Becoming an expert in such large complex systems takes years,⁸ which is why large-scale projects often have too few experts available to

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2020 The Authors. Journal of Software: Evolution and Process published by John Wiley & Sons Ltd

support the teams. Furthermore, task interdependences add to the complexity of coordination in large-scale projects. Examples of dependencies are shared resources, formal requirements and schedules, task interrelationship, and workflow constraints,^{9,10} all of which need to be coordinated. Furthermore, the need for parallel work increases the level of complexity and the number of dependencies. More coordination is needed when requirements are written in parallel, and when many components are developed and tested in parallel.^{2,11} Finally, a driver for even more complexity in coordination is distribution. You seldom find all teams and stakeholders within the same building in a large-scale project, which creates additional coordination challenges.⁶ For all these reasons, effective coordination mechanisms is one of the keys to success in large-scale software development.

Software companies have used different approaches to address coordination challenges.¹² One approach is to focus on employing skilled engineers and on improving institutionalized knowledge and technology-based knowledge repositories.¹³ Yet, keeping up active participation in maintaining up-to-date documentation is a constant struggle,¹⁴ especially so in large-scale projects. Even then, not everything can be kept in knowledge repositories, as it is hard to codify “know-hows” (e.g., tacit knowledge).¹⁵ Another approach to coordination is to focus on interactions between people or networking.^{6,15,16} However, similarly to becoming an expert, establishing a coordination network requires effort and time, as understanding who knows what and establishing connections with the right people in large-scale projects is a challenge.¹⁷ Mechanisms for cultivating social networks in large-scale projects are also not fully understood.^{18,19} Finally, to address coordination challenges, software companies have tried to limit task interdependence with approaches such as centralization and modularization.⁹ The centralization approach prescribes that certain functions such as communication with the customer,³ system testing, or system integration^{20,21} are centralized and moved out of development teams. However, this creates new dependencies and requires software teams to synchronize their work, including the exchange of feedback, with the supporting functions and roles. In the modularization approach, the software is split into modules. Modules are said to interact through well-defined interfaces, limiting work interdependences. However, software modules are rarely isolated, and therefore, maintaining and extending the modules can be difficult.²² Work interdependences can also be addressed by coordination through interactions between the right people or interpersonal coordination.⁹

Researchers emphasize that interteam coordination and coordination between teams and support roles in large-scale software development projects are challenging,^{3,23} especially in distributed projects⁶ because of organizational and temporal barriers and a lack of awareness of each other's work.²⁴ Besides, while many different approaches have emerged, none of them used single-handedly seems to be enough to solve coordination problems in large-scale software development projects.

In this paper, we contribute with further insights into team coordination in large-scale software development projects. In our study, we understand coordination as defined by Malone and Crowston, *the management of dependencies among task activities*.²⁶ We do know that in a large-scale project, teams need to coordinate dependencies with other teams, experts, and supporting roles in the project, which is known as team-external coordination.²⁷ We also know that development teams are very different and receive very different tasks,^{17,28} but we do not know much about how different teams are affected by project-, team-, and task-related characteristics, as well as impact on their teams' coordination needs. We also do not know the extent to which team-external coordination can address teams' coordination needs and to what extent team-external coordination is needed to help teams perform in large-scale software development projects.^{12,29} A better understanding of teams' coordination needs could help to decide what effort teams should invest in team-external coordination. Therefore, in this paper, we focus on teams' needs and team-external coordination in large-scale software projects and the impact of satisfaction of needs on the performance of the team (i.e., team productivity).³⁰

1.1 | Research problem

The purpose of this study is to understand the impact of team-external coordination in large-scale software development projects. To identify effective coordination practices, (1) we focus on the tasks software development teams receive, (2) how tasks affect teams' needs for coordination, and (3) what is coordinated by software development teams with other teams, experts, and support roles in large-scale software development projects. We also explore how team-external coordination impacts team performance. Therefore, our unit of analysis is a software development team in a large-scale software development project.

Our preliminary analysis demonstrated that team-external coordination differs significantly between teams.¹⁷ Therefore, the next step is to understand why team-external coordination differs, for example, what circumstances and characteristics impact the need for team-external coordination. We know that software development is knowledge-intensive work, but in which circumstances do software development teams in large-scale software development projects need to seek knowledge from other teams and experts to complete their tasks (expertise coordination)? What are the characteristics of projects, tasks, and a team that impact the need for expertise coordination? Similarly, we know that tasks are interrelated in large-scale software development, but when do software development teams need to coordinate work dependencies with other teams and support roles in large-scale development? What are the characteristics of projects, tasks, and a team that impact the need for work coordination? These topics, expertise, and work coordination, more specifically, mechanisms for team-to-team and team-to-supporting roles

coordination¹⁸ have been recognized as a high priority topic in research on large-scale software development.^{18,19} Thus, our first research question is as follows:

RQ1: What characterizes teams' coordination needs in large-scale software development?

In practice, coordination takes time and effort. Therefore, software companies are interested in what counts as effective coordination, for example, when coordination helps to increase performance instead of taking time away from coding. Coordination between teams, experts, and support roles is gaining more attention in the research community.¹⁹ However, little is known about the interplay between team-external coordination and teams' performance. One way to understand team-external coordination is to look at the content of coordination. Do teams address expertise and work coordination dependencies, and to what extent? If the expertise and work coordination needs are low, would the overly active networking influence teams' performance? If the expertise and work coordination needs are high, would the under-satisfaction of the teams' coordination needs influence teams' performance? Therefore, our second research question is as follows:

RQ2: What is the relationship between team-external coordination and teams' performance in large-scale software development?

In this paper, we used an exploratory approach to understand the factors that impact teams' coordination needs, how teams seek expertise outside of the team, and how teams coordinate work dependencies with other teams and supporting roles. We selected two case projects where we had broad access to software development teams and where we were able to gather data about teams, their daily work, team-external coordination, and the content of coordination exchanges.

The rest of the paper is organized as follows: In Section 2, we present the theoretical background and related work on expertise and work coordination in software development. In Section 3, our study methodology and overall approach are described in detail, and the empirical cases under study are outlined, providing the contextual information of the studied companies, the teams, and their tasks. In Section 4, we present our results, which are discussed in Section 5. Section 6 includes a summary of our main findings, the limitations of our study, and plans for future work.

2 | RELATED WORK

In this section, we start by defining coordination. Next, we review research on different coordination approaches, focusing on the success of these approaches in large-scale software development. Next, we review related work of expertise coordination and work coordination and respective coordination approaches. There we also review how project-, team-, and task-related characteristics can influence coordination needs and if characteristics have been reported to impact teams' performance.

2.1 | Coordination in large-scale software development projects

Coordination is needed to address dependencies among activities performed in an organization.²⁶ In this work, we rely on the definition refined by Malone and Crowston—*coordination is managing dependencies between activities*²⁶—to identify and categorize dependencies^{31,32} between activities in large-scale software development projects (see Table 1 for summary of definitions). In software systems, dependencies between different modules or services must certainly be managed, and, as teams are responsible for these services or modules, solving a task with dependencies means that teams need to interact. Furthermore, if there are no interdependencies, there is nothing to coordinate and no need for interaction. A software development team needs to coordinate constraints of, for example, requirements, testing, and integration, working together with requirement engineers, other teams, testers, and other support roles. Furthermore, one of the most critical resources for software development teams is expertise.¹⁶ A team needs to coordinate with external experts (e.g., an architect) when solving tasks (i.e., expertise coordination). Malone and Crowston's theory of coordination is useful as an analytical tool,³³ and we apply this theory as our analytical framework. According to Malone and Crowston,³⁴ communication tools are important to manage the interdependencies between tasks. Components of communication are

TABLE 1 Definitions of terms

Term	Description
Coordination	Managing dependencies between activities. ²⁶
Expertise coordination	Interactions aimed at knowing where expertise is located, knowing where expertise is needed and bringing in the necessary expertise at the right time when working on tasks in software development projects. ¹⁶
Work coordination	Interactions aimed at managing the alignment and integration of tasks, addressing dependencies that are inevitably created among teams working on software development projects. ^{2,9}
Team-external coordination	Coordination of dependencies with other teams, experts, and supporting roles in the project. ²⁷

senders, receivers, messages, and languages. Examples of these components are team decisions that require members of the team to communicate in some form about the goals to be achieved, the alternatives being considered, the evaluations of these alternatives, and the choices that are made. Such communication requires some form of “message” to be transported from senders to receivers in a language that is understandable to both. Establishing a common language for actors to communicate is therefore necessary. A description of important terms and concepts in the domain and a description of practices used in a development process are examples. Furthermore, unplanned decentralized communication is essential for rapid responses when handling interdependencies. In software teams, decentralized communication means that team members communicate directly with other team members on other teams without having to communicate through their managers. The need for decentralization is related to organizational control because decentralized communication implies decentralized control, and the benefit is faster responses to handling. Coordination is also influenced by culture (national and organizational), influences how coordination is performed. However, we do not study culture in our work.

Large-scale software development projects present some unique challenges to coordination that make them different from other software development projects. Effective coordination in software development was recognized as critical for successful software development several decades ago³ and continues to be an important topic today.¹⁸ Effective coordination in software engineering has been shown to be beneficial for project outcomes in different settings.³⁵ Task interdependencies, code complexity, and accuracy of documentation are areas of significant concern.^{9,36,37} Previous empirical research established that coordination is easier, and there is a lower need for coordination when requirements are clear and stable,³⁸ when systems are maintained, with a good refactored architecture, and well documented.³⁹ It has been shown that a shared understanding of who knows what increases performance for small, self-managing, and co-located teams.^{9,16} Most research on coordination in software development has focused on intrateam (within-team) coordination, finding that it is important that teams have shared knowledge on the tasks and how to do them, as well as who knows what in the team.^{9,16,40} We know that in large-scale projects, coordination is more difficult for teams as it can take longer to find and become aware of the right people to talk to and get an answer from.⁴ Similarly, in multisite development increases complexity of communication and coordination and cross-site work takes longer to complete⁴¹ and have negative impact on team performance.²⁹ At the same time, networking had a positive influence on the effectiveness of global software development projects.⁴² Previous research has shown that formal coordination provides some benefits for task outcomes,³⁵ and more communication helped to make interactions between teams better.⁴³ Research on interteam (between teams) coordination^{12,35} has focused on vertical structures of coordination.^{4,44,45} As a response to this, software companies have tried to create scheduled meeting structures.^{44,46} However, research about formal meetings found that they often ended up more about reporting status instead of coordinating of expertise and work dependencies.⁴⁵ Other studies focused on mutual adjustment between individuals or through scheduled or unscheduled meetings.^{46,47} Researchers have found that interpersonal coordination is important to coordinate the expertise and the workflow dependencies in the development process^{28,48} as personal knowledge often rely on personal contacts to coordinate dependencies and improve team performance.⁴⁹ However, little is known about how teams coordinate, not only within or between teams but also with experts and supporting roles and what dependencies exist between teams and supporting roles.¹² Previous research has suggested that coordination problems can be identified by investigating mismatch between dependencies and networking patterns.⁵⁰ In summary, companies working with large-scale software development projects are still challenged to find the right coordination approach to address teams' coordination needs.¹² To better understand effective coordination in large-scale software development projects, companies need to identify and categorize teams' coordination needs in relation to different project-, team-, and task-related characteristics.

2.2 | Expertise coordination

Expertise coordination is interactions aimed at knowing where expertise is located, knowing where expertise is needed, and bringing in the necessary expertise at the right time when working on tasks in software development projects.¹⁶ For example, a software development team needs to interact with other teams, an external expert, or an architect to locate and retrieve needed expertise. Furthermore, solving tasks in software development projects means that experts often need to work together. That is, a team needs to coordinate expertise when solving tasks. We know large-scale software development projects are associated with the need to possess an enormous amount of knowledge.^{3,45} Furthermore, many large systems are built on monolithic architectures. When new features or changes need to be implemented, they are rarely isolated, which increases the teams' need to know even more about the system.¹⁷ Enormous demands of expertise from development teams in large-scale projects have resulted in a significant amount of research aimed at understanding the factors that impact individual and team performance. We know that mature engineers outperform novices,⁵¹ and we know what expertise is brought to the task and that coordination effectiveness will make a difference in the project outcomes.²⁹ However, we also know that it can take years for individual developers to obtain the necessary knowledge⁵² in large-scale projects and that a single individual (or even a team) can often have gaps in understanding the entire system.^{17,36} Sharing expertise between developers is a well-known challenge in any software development project,^{15,53} and distributed development makes it even more complicated.⁵⁴ At the same time, effective expertise coordination within a software development team is linked to increases in team performance.¹⁶ However, we also know that large-scale projects have additional demands for expertise coordination with many supporting roles (for example,

system architects, system managers, system testers, and area experts) within or even across organizations in the case of distributed outsourcing projects.⁵³

Expertise coordination strategies in software organizations vary from those that focus on building organizational capital (for example, documentation) to those that focus on cultivating social capital (for example, communities of practice and other meeting forums).¹⁵ The former is reflected in the technocratic knowledge management studies that emphasize the importance of institutionalized knowledge and codified experiences.⁵⁵⁻⁵⁷ The latter are studies on behavioral knowledge management that focus on the importance of interactions between people and networking.^{9,55,58-60} However, relying only on technocratic strategies is insufficient in large-scale software development projects because documentation gets outdated fast.¹⁴ Large-scale software projects have used organizational strategies to address expertise coordination, for example, by implementing different team setups. Large-scale projects often employ either specialized teams working on certain parts of the product (component teams), universal teams working across different parts of the product (feature teams), functional (also referred to as disciplinary) teams that work only on certain types of tasks (testing teams, development teams), or cross-functional (or cross-disciplinary) teams that assume more responsibilities to implement an often end-to-end task.^{1,5} When teams specialize in fewer functions or components, these teams might lack overall domain knowledge about the product to address dependencies within the system.⁶¹ Feature or cross-functional teams might lack expertise in other parts of the system to create well-designed task solutions in the part of the system they are working with and address dependencies with other parts of the system.^{17,61} In summary, different strategies for project organization and team setups can satisfy some, but not all, expertise coordination needs in large-scale projects. As another approach, some behavioral strategies emphasize expertise coordination by networking, especially to share tacit knowledge.⁶⁰

Previous empirical research has shown that there are several project-, team-, and task-related characteristics that impact the expertise coordination needs. Some studies have focused on the experience that is needed to complete a task successfully.^{62,63} Teams' unfamiliarity with the task^{9,17} increases the need for external expertise and decreases performance. Teams' familiarity with a task is impacted by factors such as (i) team member experience, (ii) team joint work experience, and (iii) availability of product documentation. Individual team member experience is associated with better team performance.⁶⁴ Team joint work experience is mentioned in the context that team members need to be aware of the knowledge of the other team members to collectively apply this knowledge to a task.^{16,40,65} In the case of a lack of existing expertise, product documentation is one of the key knowledge resources.^{17,66} However, not every software development task requires the same amount of expertise. Task complexity is important to consider when discussing how much knowledge a team needs to possess when solving a task.^{9,16} Task interdependence is a factor that affects task complexity. When tasks are broken down to reflect the architecture or individual subsystems, teams might need additional knowledge about other subsystems and architectural layers, leading to requiring more coordination.^{67,68} Coordination, in this case, concerns the actual work coordination (as discussed below) and expertise coordination, which means that the team should understand another team's subsystem better to implement a solution that will satisfy the integration requirements. In summary, previous research has linked several team- and task-related factors to an increase or decrease in the needs for expertise coordination in software development teams and the impact of expertise possession and coordination on team performance. However, it is not clear how each of these characteristics contributes to teams' needs for expertise coordination in large-scale software development projects and whether a team can address these needs by increasing team-external coordination.

2.3 | Work coordination

Work coordination concerns interactions aimed at managing the alignment and integration of tasks, addressing dependencies that are inevitably created among teams working in software development projects.^{2,9} Tasks in software development projects require work coordination when the team needs to integrate their effort with others, and the output of another is needed as input to do their tasks.⁶⁹ Solving a task with dependencies means that teams need to interact with others outside the team to coordinate a variety of constraints. In large-scale projects, when hundreds of developers work in parallel, their actions require careful orchestration.^{1,10} At the same time, distribution, which often is the case in large-scale development, also affects coordination between teams, experts, and support roles, creating fewer opportunities for interactions.⁵ A common practice in large-scale software development projects is that work dependency coordination is performed by several coordinating roles and supported by standardizing the processes and ways of working.⁷⁰

Work coordination strategies in software organizations vary from those that focus on limiting task interdependence to those that focus on interactions between people. Impersonal coordination enabled by software architecture can be an important means to minimize the need for work coordination in software projects.^{5,67,71} However, both modularization and standardization of work, especially in distributed and global contexts, have been recognized as either insufficient³⁹ or impossible to achieve.^{5,37,72} The impact of coordination schemes, such as centralization or decentralization, has been studied, showing that coordination practices have significant influences on workflow coordination between teams.²⁸ Similarly, some coordination studies focus on means of work coordination.^{73,74} Impersonal mode coordination prescribes and focuses on standardization, consisting of formalized rules, standardization, and policies and procedures, that is, the organizational strategies.^{28,73} Personal mode coordination focuses on coordination by feedback and is enabled through mutual adjustment.^{28,73} Coordination modes were previously used to study work coordination in large-scale software development,⁴⁷ and it was found that both scheduled and unscheduled meetings are

important for coordination and that needs of work coordination change over time. Large-scale software development projects have been traditionally associated with plan-driven management and coordination by standardization.^{20,21} However, empirical studies in large-scale software development suggest that the success of standardization and modularization is limited.³⁹ Considering these findings, networking as a mechanism of work coordination has been identified to be important.^{4,9,12,17,75,76}

Previous empirical research has established that there are several project-, team-, and task-related characteristics that impact work coordination needs. In software development, each development team needs to be familiar with the processes required to integrate their software development task in the final product successfully. Therefore, unfamiliarity with the processes in the project will increase teams' need for external work coordination, and teams will need to put effort into becoming aware of and addressing different process dependencies related to their tasks.⁷⁶ Factors such as (i) process stability and (ii) process documentation can ensure that people stay familiar with a process or learn about the process from readily available sources. When the process is stable, teams gain an understanding of ways of working on the project over time and require less effort to figure out and address task dependencies.¹⁷ Sufficient process documentation on how task-related work should be done and well-documented ways of working will decrease the need for a team to seek this information in the rest of the organization.⁶⁶ Process complexity will also affect the need for work coordination. Process complexity can originate from (i) process dependencies and (ii) task dependencies. Process dependencies increase when certain development functions are placed outside of a team due to, for example, process centralization in large-scale software development projects. When that happens, teams have to rely on others.⁶⁷ Work dependencies in large-scale software development, for example, are the need to approve design proposals with architects, handshake on solutions with experts, or require external code reviews.⁷⁷ There is also an increase in work coordination needs when software development teams work in parallel with other teams on different software development tasks.¹ Process complexity can be further impacted by task interdependence, as tasks might have more dependencies with other subsystems and layers.⁵ In summary, in previous research, several factors have been individually linked to an increase or decrease in the needs for work coordination in software development teams and how work coordination impacts team performance. However, as in the case of expertise coordination, it is also not clear how each of these characteristics contributes to teams' needs for work coordination in large-scale software development projects and whether a team can address these needs by increasing team-external coordination.

3 | METHODOLOGY

In this section, we describe the set of criteria for selection of the cases and describe characteristics of both cases we selected. Further, we describe data collection methods: interviews, focus groups and questionnaire and how data were collected. Lastly, we describe data analysis methods to determinate teams' coordination needs and visualize team-external coordination patterns and analyze content of reported coordination connections.

3.1 | Empirical cases

We aim to learn about and to better understand factors that impact team needs, team-external coordination, and performance in a complex environment—large-scale software development. To understand the characteristics that impact teams' needs and the relationship between the satisfaction of these needs by team-external coordination and team performance, we used an exploratory approach.⁷⁸ We conducted an embedded multicase study in which the unit of analysis was a software development team in a large-scale distributed software development project.⁷⁸ In our exploratory research, we relied on theoretical sampling to select cases. Cases were selected based on following criteria: (a) large-scale project, (b) distributed development effort, (c) that develops knowledge-intensive software solution, (d) offshore-inhouse software development project, and (e) company is based in Sweden. This allowed us to select two projects in companies that we had access to Project A and Project B and study nine selected software development teams. In both organizations, we selected teams from each of the main development locations. Development teams with overlapping and complementary characteristics were selected. We selected teams that are both mature as well as relatively new teams and teams that worked on tasks with different task complexity and interdependence. In Project A, all selected teams were feature teams. In Project B, all selected teams were component teams. This sampling was done with the help of company representatives.

3.1.1 | Case: Project A

The first case is from company that develops generic software products offered to an open market and complex compound systems with customized versions. At the time of our investigation, Company A had worked in agile development for almost 6 years and at present, agile practices have become standard practice. The investigated project in Company A is the development and maintenance of a subsystem that has multiple components and interfaces with other subsystems. The subsystem is very complex and contains several million lines of code. Eight years prior to our study, the company realized that it lacked competence in certain areas and experienced problems in the implementation of new features in the large-scale system. The size and complexity of the system domain and product knowledge are such that several years are required to become a

knowledgeable developer. This resulted in a shift towards cross-functional development teams that are responsible for a feature from high-level description until its release to the customer. Setup of Project A is offshore-inhouse software development. The number of developers working on the subsystem grew from eight in 2007 to five teams in Sweden, eight in-house teams and two offshore outsourcing consultant teams in China, and two Korean teams. Cross-functional feature development teams were supported by additional roles for program management, product management, and release management. Development work at the time of our investigation in early 2014 was organized in 17 self-managing cross-functional feature teams.

3.1.2 | Case: Project B

The second case is from company that develops automated software solutions and embedded software for various industries. The investigated project in Company B was a complex system with multiple modules with coupled dependencies and requires integration of expertise about various areas and functionality. The system contains several million lines of code and is written in multiple programming languages. Company B places a large emphasis on product and process quality, with one of the reasons being to meet safety-critical requirements. The project follows a V-model development methodology. Tasks are structured in modules, and component teams are assigned to these modules. Because of the size and complexity of the system, a good developer has an overview of half of the system. Setup of Project B is offshore-inhouse software development. In early 2014, the work was distributed across two sites in Sweden and a site in India, all branches of Company B. There were six software teams in *Sweden 1*, a software team in the *Sweden 2* site, and a group of developers in India. Both of the Swedish sites were engaged in system development from the onset, while the Indian site joined in 2006. The main Swedish site, referred to as *Sweden 1* in the results, is responsible for the product and employs the majority of software developers. As such, the software projects are usually led from this location with subproject managers in each location. Hardware expertise resides in the second Swedish location (referred to in the results as *Sweden 2*), and thus, hardware projects are led from this location. Development work is organized in projects, which can include the development of new generations of the system, new functionality development, roll-ups of large maintenance projects, and pure maintenance projects.

3.2 | Data collection

Qualitative data from the multicase study were collected based on 16 interviews, nine focus groups, and observations from visiting five different sites in the two companies. Quantitative data were collected using a questionnaire that facilitated the collection of 49 responses in total (see Table 2).

3.2.1 | Semi-structured interviews

Interviews with individual members were organized for each of the companies. These interviews followed a semi-structured agenda, where individual members of different management levels were asked directly about the following: (i) project characteristics to clarify the type and complexity of the tasks in the investigated projects; (ii) their involvement in the project; and (iii) organizational process and connections that are necessary for teams when working on their tasks. In the interviews, we also asked project managers to describe the daily tasks performed by the teams that participated in focus groups. All interviews in both companies were performed in English, recorded, and later transcribed.

3.2.2 | Focus groups

After the interviews, focus groups with selected development team(s) members were organized in each of the companies. Focus groups is a research technique⁷⁹ that collects data on research themes through group interaction. Focus groups is a group interview based on interaction within the group of practitioners, based on topics that are supplied by the researcher, typically taking the role as a moderator. The explicit use of group interactions produces data and insight that would be less accessible without the interactions found in a group. This means that the researchers should have an active role in creating the group discussion and the iteration in the group as the source of data.⁷⁹ The focus groups followed a structured agenda, facilitated by the researcher. A portion of the time was dedicated to acquiring information about each team. Themes were decided beforehand and derived based on research objectives. We asked each team about (i) the types of tasks they receive and the task characteristics such as familiarity, (ii) the presence of the skills needed to address the team's tasks, (iii) teamwork practices, (iv) interaction with other teams, roles, and communities, (v) teams reliance on different sources of knowledge in their daily work, the teams' perception of their performance, and (vi) the perceived usefulness of the team-external coordination. Focus groups were organized with nine selected development

TABLE 2 Data collection

Company A						
Activity	Location	Time	Participants	Researchers	Data gathered	
Questionnaire	Sweden	Apr 2014	3 development teams, 16 respondents	A1, A2, A3	Individual connections, purpose of communication, and coordination.	
	China	Mar 2014	3 development teams, 19 respondents	A1, A2, A3		
Interviews	Sweden	Aug 2013	7 interviews	A2, A3	Description of the project, task assignments to teams, task characteristics.	
	Sweden	Oct 2013	3 development teams	A2, A3	Presence of expertise in teams. Reliance on different	
Focus groups	China	Mar 2014	3 development teams	A2, A3	sources for expertise and coordination and external interactions with other members in project.	
	Sweden	Oct 2013	3 development teams	A2, A3	Presence of expertise in teams. Reliance on different	
Company B						
Activity	Location	Time	Participants	Researchers	Data gathered	
Questionnaire	Sweden1	Feb 2014	2 development teams, 9 respondents	A1, A2, A3	Individual connections, purpose of communication, and coordination.	
	Sweden2	Feb 2014	1 development teams, 5 respondents	A1, A2, A3		
Interviews	Sweden1	Aug 2013	6 interviews	A2, A3	Description of the project, task assignments to teams, task characteristics.	
	Sweden2	Feb 2014	3 interviews	A1, A2	Presence of expertise in teams. Reliance on different	
Focus groups	Sweden1	Feb 2014	2 development teams	A1, A2	sources for expertise and coordination and external interactions with other members in project.	
	Sweden2	Mar 2014	One team in the other Swedish location	A1, A2	sources for expertise and coordination and external interactions with other members in project.	

*This column refers to the authors' participation in the data collection; the first author is A1, the second author is A2, and third author is A3

teams in total. We organized one focus group per team. The number of team in each of the company and total number of participants in focus groups are reported in Figure 2. In each focus group, the majority of the team members participated; however, some were unable to attend the focus groups as they were involved in training or on leave. The activities of all focus groups in both companies were recorded and later transcribed by the researchers.

3.2.3 | Questionnaire

A questionnaire was used and focused on capturing coordination contacts, that is, important connections of each team member from the same nine teams that participated in focus groups. Also, the participants reported the content of each connection to their contacts. The participants reported their contacts in a free-recall format. We decided to use a questionnaire because it is a common method for collecting empirical data in the social sciences and also is recognized in software engineering⁸⁰ as a means to systematically collect relation data.⁸¹ We conducted a questionnaire with the same nine teams that participated in the focus groups. The questionnaire was web-based for the Company A Swedish site, whereas paper printouts were used for the Company A offshore site, two Company B Swedish sites. A web-based questionnaire for the Company A site was sent a few months later, but all other questionnaires were administered immediately after the focus groups. We partially replicated a questionnaire conducted by Manteli et al.⁸² All respondents were asked to report their answer to the question "What kind of knowledge you transfer/receive from this person" with each identified contact in free text format. For each identified contact, they also provided an answer on the frequency of coordination with contact using the 5-point Likert scale (from *Never to Sometimes to Daily*). Using this format, we collected team coordination contacts using a "realist" approach. By using this approach, we rely on the network that are perceived by the participants will correspond to the actual boundaries of social groups and organizations.⁸³ Participants in the paper-based questionnaires were given approximately 30 min to fill in as many responses to the questionnaire as they determined to be necessary in a "free-recall" format. Participants in the web-based

questionnaire were given 2 days to fill in as many responses as determined to be necessary, also in a "free-recall" format. A longer time was required for the online process because of the need of the participants to allocate time to this activity in their daily work schedule. We completed the list of recalled contacts and clarified these with company representatives to verify the names, roles in the company, and location. The size of the participating teams ranged from 5 to 9. Not every member of each team completed the questionnaire, resulting in a usable sample of 49 members and nine teams (see Table 3). (1) Company A: In total, 35 people from the six teams completed the questionnaire. The response rate for the six teams in total was 90%, and for individual teams, this rate did not fall below 71%. (2) Company B: In total, 14 people from three teams completed the questionnaire. The response rate for the three teams in total was 88%, and for individual teams, the rate did not fall below 67%.

3.2.4 | Performance measurements

Performance measures are an important consideration in group research. In a study of software development teams,¹⁶ Faraj and Sproull found that expertise coordination within teams affected some performance measures but not others. We follow a similar scheme and use team effectiveness as a measure for performance. Team effectiveness was measured by evaluating five items: work quality, team operations, ability to meet project goals, extent of meeting design objectives, and reputation of work excellence.⁸⁴ Following Faraj and Sproull,¹⁶ we collected performance data via subjective assessments made by the interviewees that occupied managerial roles. Although these evaluations were subjective in nature and also a threat to construct validity, we chose to use these measurements to compare performance across teams and companies, because objective measures, such as lines of code per person month or team velocity, might not be comparable across different companies and, thus, are often subject to misinterpretation. All items for team effectiveness were measured using a 5-point scale.

For each team included in the study, we collected performance measurements of team efficiency (five items). One response per team was collected from their respective line managers for the six teams in Company A and one response per team from respective line managers for the three teams in Company B. A summary of the collected performance measurements is presented in Table 6.

3.3 | Data analysis

Our data analysis followed a two-step process. The first step involved identifying and quantifying team task characteristics using qualitative analysis performed by coding the interview and focus group transcripts. Our approach entailed within-case and team-by-team analysis. The second step involved analyzing the questionnaire data in which we coded the qualitative part of the responses (the content of coordination connections) and then analyzed team-external coordination using social network analysis techniques.

3.3.1 | Analysis of teams' coordination needs

We determined team task characteristics by analyzing the data obtained: interview transcripts, focus group transcripts, and collected qualitative information about the teams based on the questionnaire. Transcriptions of interviews and focus groups and coding were done by first and second author, using NVivo software. We started to analyze teams' coordination needs using provisional coding process, where a predetermined list of codes based on previous research was applied to the data,⁸⁵ as illustrated in Figure 1. Specifically, we used an approach identified by

TABLE 3 Survey participants

	Teams	Location	Team members	Participated	Response rate
Project A	Team A	Sweden	8	6	75%
	Team B	Sweden	7	5	71%
	Team P	Sweden	6	5	83%
	Team G	China	9	8	89%
	Team N	China	5	5	100%
	Team R	China	6	6	100%
Project B	Team I	Sweden	5	5	100%
	Team C	Sweden	6	4	67%
	Team S	Sweden	5	5	100%
	Total:		57	49	86%

DeCuir-Gunby et al.⁸⁶ to develop predetermined *theory-driven codes* as the strategy for the development of initial codes and coding. We determined theory-driven codes with our literature review of factors (based on Related Work Section 2.2 and Section 2.3) that influence teams' coordination needs in two categories: (i) expertise coordination and (ii) work coordination. In this literature review, we focused on reviewing factors and task characteristics that have been reported as having an effect on team performance. In Table 4, we report these factors and related effect (positive or negative). As a result of coding process, we identified the values of each of the factors for each team, as illustrated in Step 2 in Figure 1. In the second step, we used magnitude coding. Magnitude coding is a coding procedure to associate intensity or frequency to an existing code datum or category⁸⁵. This process is used to quantify the extent of the individual teams task characteristic factors and mediating factors, to then compute the overall extent of the task characteristic factors. When computing the final magnitude of the task characteristics, we added our interpretation of the impact of each factor (task characteristics and mediators) for each team, as illustrated in Step 3 in Figure 1. The results of the coding process are shown in Table 8 (expertise coordination needs) and in Table 9 (work coordination needs).

Triangulation of the data sources was performed to increase the reliability of the resulting values. For example, we measured team member experience in two ways—by averaging each team member experience in the organization and by using the perceived level of technical skills and product knowledge in relation to team tasks, as discussed during the focus groups. Similar to the performance data, we used the 5-point Likert scale to express the magnitude of each factor (from *Very Low* to *Moderate* to *Very High*). In a few cases, we were not able to determine the level of a factor because of missing or limited data from the focus groups (identified with a question mark).

3.3.2 | Classification of team-external coordination contacts

The analysis of connection classification followed a three-step process. First, the qualitative data that described the content of coordination connections from the questionnaire were coded and aggregated using the bottom-up approach. We piloted analysis by following the Provisional

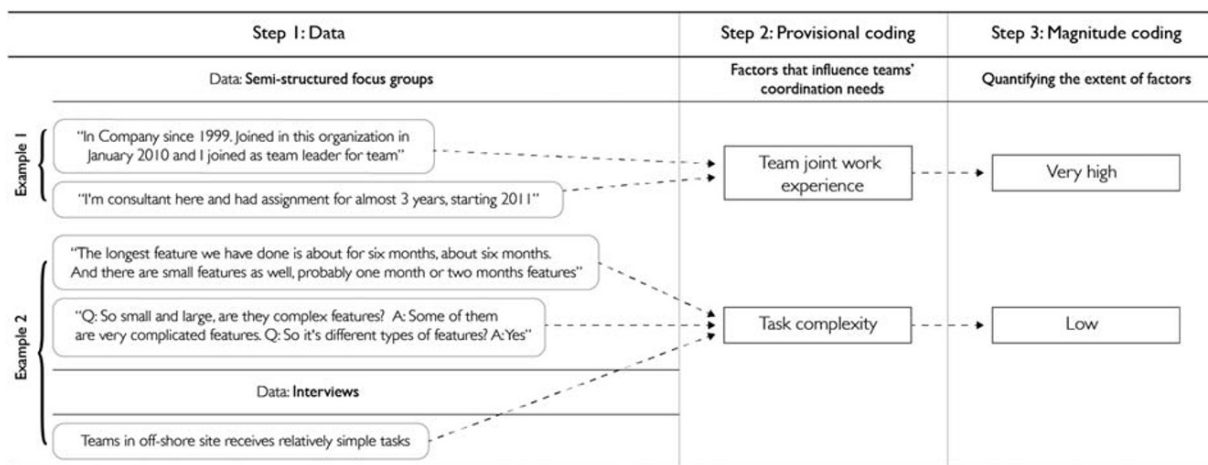


FIGURE 1 Illustration of coding process to analyze teams' coordination needs

TABLE 4 Summary of characteristics impacting teams' coordination needs

Expertise coordination		Work coordination	
Factor	Impact	Factor	Impact
Task familiarity	Decreases	Process familiarity	Decreases
Mediated by:		Mediated by:	
Team member experience	Decreases	Process stability	Decreases
Team joint work experience	Decreases	Process documentation	Decreases
Product documentation	Decreases		
Task complexity	Increases	Process complexity	Increases
Mediated by:		Mediated by:	
Task interdependency	Increases	Task interdependency	Increases
		Process dependency	Increases

Coding approach.⁸⁵ However, our coding soon revealed that the predetermined provisional codes based on the relevant literature review (Section 2) were insufficient. Therefore, we created a new iteration of coding and identified new common themes the content of coordination connections and refined the codes. Using refined codes, we classified each reported coordination connection into two separate categories: expertise coordination and work coordination, as illustrated in Figure 2. The result of this step is the classification of the content of coordination connections (see Table 5).

In the second step, we aggregated connections, to create a set of contacts for expertise coordination and work coordination for each team in the study. We used unique contacts per team measure because we focused on team-external coordination in general, not individual team members.^{17,77} As such, this measure better represents the effective team-external coordination. We distinguished between contacts and connections in the following way: Connections indicate all reported connections by each team member, while contacts are unique contacts per team (e.g., if two team members report the same individual in their connections, this is counted as only one unique contact).

In the final step, we depicted team-external coordination. Team-external coordination patterns were visualized using the Fruchterman-Reingold layout algorithm.⁸⁷ The expertise coordination and work coordination are shown in different colors, and all team members are aggregated in a single node to emphasize the external coordination contacts.

3.3.3 | Team performance

In our analysis, we aimed to achieve a single score for the team performance based on the aggregated team effectiveness score.¹⁶ First, we averaged the five items to develop a measure of team effectiveness (mean = 4.07, SD = 0.75). A summary of the performance measurements is presented in Table 6. To test for internal consistency of the performance measurements, we calculated the Cronbach alpha levels of all the variables. The Cronbach alpha level of effectiveness is .86 on the team level. These values are above threshold level .7 recommended to constitute a good level of internal consistency^{88,89} and also are in line with the levels achieved in the source of team performance measurement.¹⁶

4 | RESULTS

In this section, we first describe each team and their networks for two categories: expertise coordination and work coordination. Second, we report on the teams' coordination needs for expertise and work coordination. Finally, we demonstrate the relationship between team-external coordination and teams' performance.

4.1 | Team-external coordination patterns

We summarized coordination contacts (unique team-external coordination contacts for a team) and coordination connections (total reported connections from all team members for a team). In Table 7, we report exhibited team coordination data based on the analysis of questionnaire data for two predetermined categories: (1) expertise coordination and (2) work coordination. It was determined that expertise coordination contacts per team were between 4 and 20 contacts (mean = 9.78, SD = 4.87) and 4–28 connections (mean = 14.22, SD = 6.48) per team. This shows relatively large overlapping connections for expertise coordination, which indicates that team members individually rely on the same experts. With respect to work coordination, it was determined that there were 2–30 contacts per team (mean = 11.78, SD = 7.74) and 2–35 connections (mean = 13.56, SD = 8.88) per team. Notably, there is less redundancy in work coordination contacts. Team R had the smallest number of team-

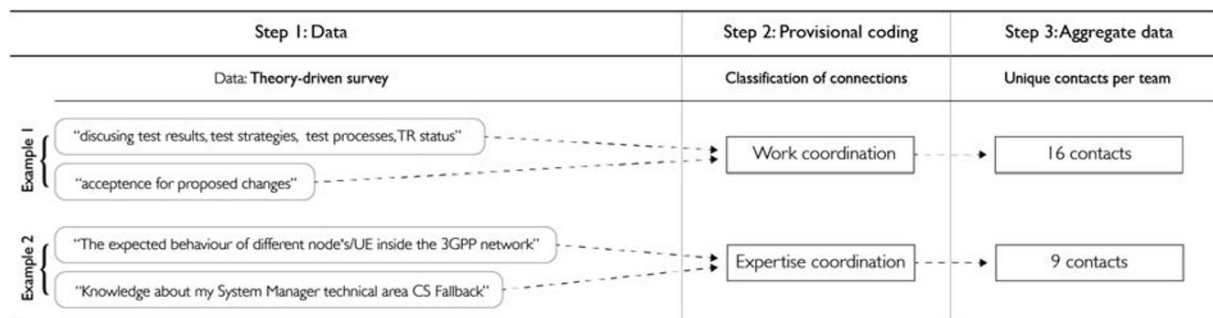


FIGURE 2 Illustration of coding process to classify team-external coordination contacts

TABLE 5 Classification scheme of reported content in the coordination contacts

	Category/subcategory	Description	Examples
Expertise coordination	Technical information	General knowledge and skills in software	Programming languages skills, testing skills,
		development.	knowledge about general security standards, general communication standards and protocols.
	Domain information	Knowledge about the application domain	Expertise gained in working in the same domain
		area and client operations.	area but not necessarily specific to certain product.
		Knowledge and skills that are transferable between systems or can be learned without working on the specific system.	
	Product information	Knowledge about specific software product	Knowledge about software product parts, standards,
and specific knowledge about the application		protocols and architecture, code properties, testing,	
domain area.		concept location within the code.	
Work coordination	Feedback on teams' output	Coordination of teams' deliverable.	Issues, feedback on testing, results of testing,
		Feedback on design proposals, code	proposals on changes, reviews, implementation
		integration in the system, and testing results.	proposals, integration issues.
	Interrelated task activities	Alignment of ways of working when multiple	Ways of working with environment/tools including
		teams and different parts of the organization are working together.	testing, ways of working with how to deliver in product, design rules, tips and tricks, alignment of team processes.
Administrative information	Instructions	Formal requirements for work organization.	What to do, schedules, deadlines, priorities from line.
	Reporting	Formal reporting to line organization.	Line progress and status updates, information about
		project milestones and delivery schedules, formal policies and procedures.	
	Personal development and methodology	Knowledge about individuals performance and methodology.	Soft-skills, knowledge of Scrum, Agile, team-building (and other processes), knowledge related to individuals' improvement.

TABLE 6 Team performance: effectiveness

Team	Company A						Company B		
	A	B	C	D	E	F	G	H	I
Work quality	4	5	4	4	4	3	5	4	4
Team operations	5	3	5	5	4	2	4	3	4
Ability to meet project goals	5	4	5	4	5	2	4	4	5
Extent of meeting design objectives	4	4	4	3	4	3	4	4	5
Reputation of work excellence	5	4	5	4	4	1	5	4	5
Effectiveness	4.6	4.0	4.6	4.0	4.2	2.2	4.4	3.8	4.6

TABLE 7 Team-external coordination data

	Team		Network contacts				Network connections		
	Name	Participated	Expertise	Size	Process	Size	Expertise	Work	
Company A	A	6	10	1.67	30	5.00	14	35	
	B	5	6	1.20	14	2.80	9	14	
	C	5	4	0.80	9	1.80	4	10	
	D	8	6	0.75	12	1.50	17	13	
	E	5	12	2.40	13	2.60	16	13	
	F	6	7	1.17	2	0.33	14	2	
Company B	G	5	13	2.60	7	1.40	13	9	
	H	5	10	2.00	8	1.60	13	12	
	I	4	20	5.00	11	2.75	28	14	
		Mean		9.78	1.95	11.78	2.20	14.22	13.56
		SD		4.87	1.32	7.74	1.31	6.48	8.88
		MIN		4	0.80	2	0.33	4	2
	MAX		20	5.00	30	5.00	28	35	

external coordination contacts with the lowest number of reported contacts for work coordination (2 contacts) and below average number of contacts for expertise coordination (7 contacts). The results show that there is a large standard deviation in the work coordination contacts and connections, indicating that some teams have to manage work dependencies with others more often.

Figure 3 shows all the graphs of individual team-external coordination patterns. We indicate expertise coordination connections and work coordination connections using different colors, and the team members are represented as one central node. The contact position in the team-external coordination graph is related to the frequency of communication—the closer to the team, the more frequent the communication. Figure 3 shows that teams exhibit different team-external coordination patterns with respect to the size, contact redundancy, and the intensity of expertise coordination versus work coordination. Teams A–F are feature teams from Case A, where Teams D–F are offshore teams. Teams G–I are component teams from Case B, where Team G is team from another Swedish location. We observed a pattern for many cases of multiple connections to single contacts, which indicates large contact redundancy (Team D and Team I). We also observed a pattern of nonoverlapping connections to a high number of contacts, which might indicate that the team is in search of specific knowledge or working on highly interdependent tasks (Team A). When comparing the intensity of expertise coordination versus work coordination, we identified differences across the two companies. In Company A, it was determined that all teams (except Team F) have more contacts for work coordination than for expertise coordination. In Company B, the opposite was determined. For example, Team A and Team D exhibited a large number of work coordination connections compared with expertise coordination connections, while Team H and Team I exhibited the opposite. The teams also revealed different redundancy in team-external coordination contacts in addition to the number of coordination contacts and connections. Team D and Team F show relatively large redundancy in connection (this means that many team members reported the same contacts outside of the team), while Team A and Team G demonstrated the opposite.

In Figure 4, we illustrate team-external coordination based on the number of contacts per respondent. The Figure 4 shows the relative differences in team-external coordination, and it can be seen that the number of contacts per respondent for expertise coordination is more even (eight out of nine teams have network size [contacts per participant] between 0.8 and 2.6), while the number of contacts per respondent for work coordination are more diverse (eight out of nine teams have network size [contacts per participant] between 0.33 and 2.8). Average sizes of the number of contacts per respondent are 1.95 (expertise coordination) and 2.20 (work coordination). Our analysis of the links between team-external coordination and performance is based on contacts per respondent measure. In summary, when size is adjusted based on the number of contacts per respondent, there is fewer differences in relative amount of team-external coordination.

4.2 | Profiles of teams and their coordination needs

Here, we report team profiles with respect teams' coordination needs in two categories: (1) expertise coordination (in Table 8) and (2) work coordination (in Table 9). These are based on the qualitative analysis of focus group transcripts during which we coded the data with pre-determined task characteristics (as described in Section 3.3.1). Qualitative data were quantified using magnitude coding (as described in Section 3.3.1).

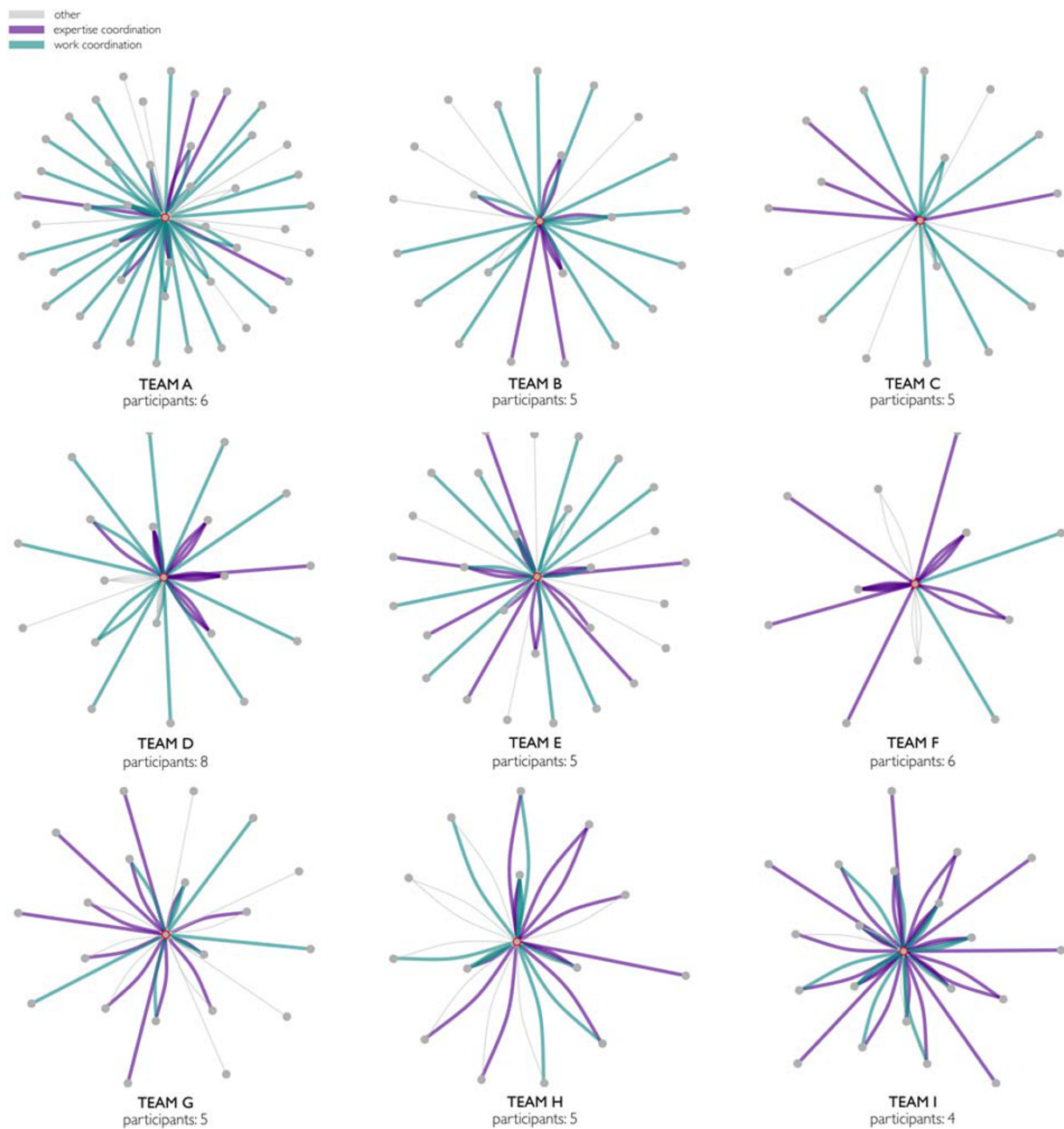


FIGURE 3 Team-external coordination patterns

In Table 8, we report factors that impact the teams' coordination needs for expertise coordination: (i) task familiarity, supplemented by three mediating factors: joint work experience, joint team experience, and product documentation; and (ii) task complexity, supplemented by one mediating factor: task interdependence. In Table 9, we report on factors that impact teams' needs for work coordination: (i) process familiarity, supplemented by two mediating factors: process stability and process documentation; and (ii) process complexity, supplemented by two mediating factors: task interdependency and process dependencies.

To understand the similarities and differences between different teams and their teams' coordination needs, in Figure 5, we plotted all teams based on two factors of task characteristics for each of two categories: (1) Needs for expertise coordination are determined by task familiarity and task complexity, and (2) needs for work coordination are determined by process familiarity and process complexity. We distinguished between four different quadrants of task characteristics (combinations of low/high in two factors of task characteristics). Based on the argument that teams in the same quadrant exhibit similar teams' needs for team-external coordination, we grouped these teams together to present a qualitative data summary of the task characteristics and their impact on teams' coordination needs. Different shapes in Figure 5 represent the affiliation to a specific group. We identified four different groups of teams in each of the two categories. In the next two sections, we describe the groups in each category, the similarities in the task characteristics within each group and differences in task characteristics between groups.

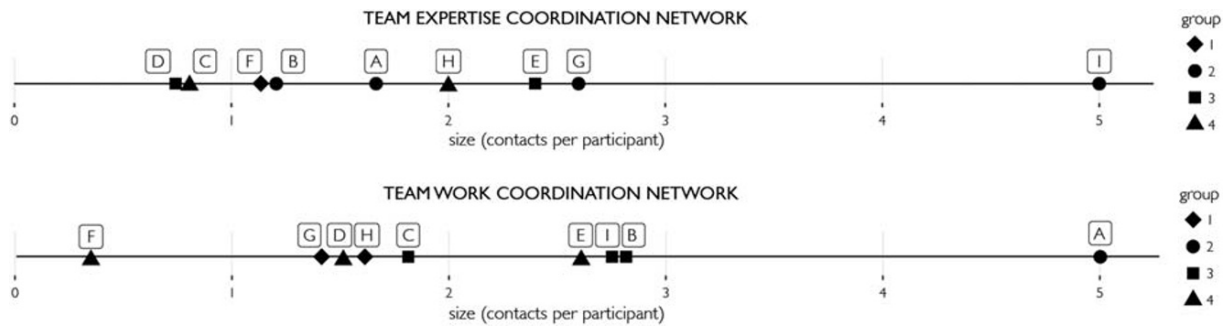


FIGURE 4 Team-external coordination patterns—contacts per respondent in the team

TABLE 8 Team-by-team analysis of needs

Team	Company A						Company B		
	A	B	C	D	E	F	G	H	I
Task familiarity	Low	Moderate	High	Moderate	Moderate	V-Low	Moderate	High	Moderate
Team member experience	V-High	V-High	V-High	Low	Low	V-Low	V-High	High	V-High
Team joint work experience	High	Moderate	High	Moderate	Moderate	V-Low	V-High	High	V-High
Product documentation	Low	High	High	High	?	High	?	Low	Low
Task complexity	High	V-High	High	Low	Moderate	V-Low	High	Moderate	High
Task interdependency	V-High	High	High	Low	Low	V-Low	Low	Low	High

Note: Need for expertise coordination.

TABLE 9 Team-by-team analysis of needs

Team	Company A						Company B		
	A	B	C	D	E	F	G	H	I
Process familiarity	Low	Moderate	Moderate	Moderate	Moderate	Low	High	High	Moderate
Process stability	Low	Low	Low	Moderate	Moderate	Moderate	High	High	High
Process documentation	Low	Low	Low	Moderate	Moderate	Moderate	High	High	High
Process complexity	V-High	High	High	Moderate	Moderate	Low	Moderate	Moderate	High
Task interdependency	V-High	High	High	Low	Low	V-Low	Low	Low	High
Process dependency	High	High	High	Moderate	Moderate	Moderate	High	High	High

Note: Need for work coordination.

4.2.1 | Teams' needs for expertise coordination

In the first of the four groups, as shown in Figure 5 (left matrix), is teams (Team C and Team H) with moderate needs for expertise coordination, and teams in this group have high task familiarity and moderate task complexity factors. In the second group is teams (Team D and Team E) with low needs for expertise coordination; these teams have tasks with moderate task complexity and high or moderate task familiarity. The third group includes Team F with moderate needs for expertise coordination. The team receives tasks with low task complexity and have low task familiarity. In the fourth group is teams (Team A, Team B, Team G, and Team I) with high needs for expertise coordination; these teams are assigned to tasks with high or very high task complexity and often are challenged working with unfamiliar tasks. To discuss groups, we follow the same structure for all groups and first report team member experience, affiliation, the presence of experts, then member “attrition” or team stability, the characteristics of the tasks, and finally, the characteristics of the team-external expertise coordination. The summary of these results is found in Table 8.

Group 1 (Team C and Team H)

Teams in this group show moderate needs for expertise coordination. The teams receive relatively familiar tasks with moderate-to-high task complexity, but the expertise coordination needs are moderate because teams in this group have a high level of competence.

Teams C and H in this group have high experience (average 17.4 and 8 years, respectively) and the high joint work experience (average affiliation with the team is 2.4 and 4.4 years, respectively). Both teams rate their existing technical competencies and product knowledge in a team as good, which means that teams generally have the expertise needed to address their tasks. Notably, both teams also have expert members. Having

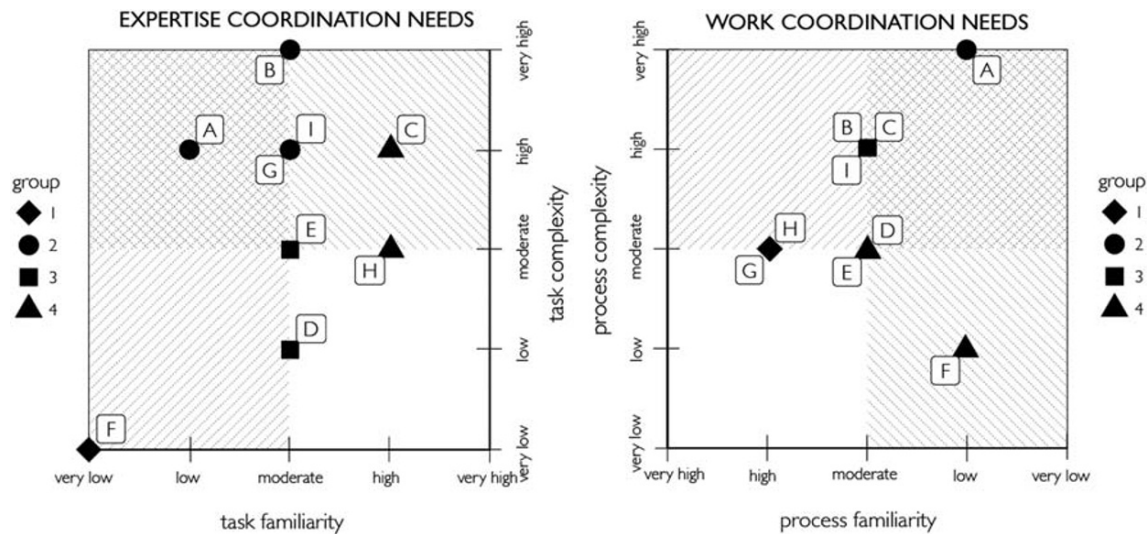


FIGURE 5 Grouping of teams based on teams' coordination needs

an expert helps to address the expertise needs in the team and also facilitates the identification of the appropriate individual to contact outside of the team if needed, as one of the experts noted: *Since you are responsible for a technical area, you always know what is going on [in the] project.* In addition, both teams either have part-time members or have encountered recent changes in team membership, as a member of team C noted: *Someone is always leaving or joining, but someone "old" always stays.* Two out of five team members in team H are relatively new to the team.

The occasional work in unfamiliar system areas and an occasional increase in task complexity are the reasons for seeking expertise outside of the team. As noted by a member of team C with respect to the variance in their tasks: *[Feature work] could be from half a year to a couple of months, and we also have trouble report sprints where we correct trouble reports for approximately three months.*

Team C has a small number of team-external expertise coordination contacts with only four reported contacts (0.80 contacts per participant), and team H has the smallest number of team-external expertise coordination contacts reported from Company B ten reported contacts (2.00 contacts per participant).

Group 2 (Team D and Team E)

Teams in this group show low needs for expertise coordination. This is because these teams are relatively experienced for the offshore site and receive tasks with low-to-moderate task complexity.

Team D and Team E have an average of only 2 and 2.8 years of experience in the project, respectively. Joint work experience in these teams is moderate (average affiliation with the team is 2 for Team D and 2.2 years for Team E). Both teams rate their existing technical competencies and knowledge as moderate (on average). However, experiences are slightly different among the members of a team: *Some of us are very strong. Some of us are very weak.* Generally, teams agree that they have the expertise needed to solve their tasks. Team E has expert members: *It [experience] is average because we have some experts in our team and they may pull up the overall averages.* In addition, team E has been one of the leading teams in the offshore location that trains newcomers. Team D does not have any formal experts in their subsystems. Both teams either have part-time members or some have joined very recently or have experienced frequent changes in team membership, as noted by a member of team E: *Many team members changed during these three years.* Member "attrition" and part-time members in these teams do not allow them to create a stable knowledge pool within the team. This increases the need to seek knowledge outside of the team, as a member from team D noted: *Although we have a lot of people [in the team] we still have resource problem in our team because some people are just part-time.*

Both teams receive relatively simple tasks, small features, and problem reports. Task complexity for teams varies and the longest features take up to 6 months, but there are also features that are as short as one or 2 months. Team E occasionally receives features with higher task complexity. Deficiencies in existing expertise in the team that was attributed to limited team stability and was one of the reasons that members sought interactions with older members and relocated to other teams until the needed experience was regained. However, team G also places a strong emphasis on teamwork, which limits their coordination with others.

Team E has reported 12 different team-external expertise coordination contacts (2.40 contacts per participant), while team D has reported six team-external expertise coordination contacts (0.75 contacts per participant).

Group 3 (Team F)

In this group (i.e., Team F), needs for expertise coordination are moderate. Although Team F receives tasks with low complexity, this team is relatively new and inexperienced and still needs to acquire more expertise to solve even those tasks.

Team member experience in the company is low for Team F (average 2.2 years, all team members are new in the organization, except for one member with 11 years of experience). The team joint work experience for team F is very low, with an average affiliation with the team of 0.9 years. The team indicated that existing technical competence and product knowledge is low and that a significant amount of additional expertise about the system is required. Team F does not have any experts. The team is new, with numerous new members in the company and therefore has not created a stable knowledge pool within the team.

Team F is the offshore team and receives relatively simple tasks, small features, and trouble reports. However, because team F is inexperienced in the project, even simple tasks create the need to acquire new task knowledge to solve them. For new and inexperienced teams, product documentation is often the source of task knowledge, and Team F perceives that the source code is a good resource and that there is sufficient documentation on the system architecture. The team noted that some of the product documentation had not been updated recently.

Team F reported a small number of team-external expertise coordination contacts with seven unique contacts (1.17 contacts per participant).

Group 4 (Team A, Team B, Team G, and Team I)

Needs for expertise coordination are high for teams in the fourth group. All teams are experienced and receive relatively unfamiliar tasks with high complexity.

The teams are very experienced in organizations (average of 10.8, 12.1, 17.2, and 11.3 years), and the joint work experience for all teams is high (average affiliation with the team is 3, 9.1, 6.1). The team members have a lot of existing technical competencies and product knowledge. However, the teams also have indicated that they need even more expertise. A developer from team I admits: *It's a very complex system. So you don't know everything about it.* Although all the teams have experienced members, only Team B has a formal expert (area architect) as one of its members. All teams in this group have been relatively stable over the last few years except for team B, as two individuals and the team leader left, and one team member works part-time.

Extensive experience is often the reason they are assigned to complex tasks, as team A member noted: *Features coming to us are more and more complex. I think that is a challenge.* Team A also receives unfamiliar tasks relative to their peer teams: *We have many senior people with very deep competence in specific sub-systems, and now they are learning other sub-systems.* Complexity of tasks requires teams to seek new expertise from experts, as one team member stated: *We are also comfortable with relying on experts in different areas.* Further, from the teams' perspectives, the quality of product documentation is mixed, such that the more common parts are good, but a lot of effort is needed to identify it, and it is not always useful. This again leads teams to rely more on discussions with the experts from different product areas.

Two teams in this group Team A and Team G reported an average number of team-external expertise coordination contacts with 10 and 13 contacts (1.67 and 2.60 contacts per participant). Team B reported six team-external expertise coordination contacts (1.20 contacts per participant), which could be due to the presence of a formal expert in the team. Team I reported the largest number of team-external expertise coordination contacts in our sample with 20 unique contacts (5.00 contacts per participant).

4.2.2 | Teams' need for work coordination

In the first of the four groups, as shown in Figure 5 (right matrix), is teams (Team G and Team H) with low needs for work coordination, as these teams are familiar with the process and work on tasks that are relatively isolated. In the second group is teams (Team D, Team E, and Team F) with moderate needs for work coordination because the tasks do not have many dependencies. In the third group, is teams (Team B, Team C, and Team I) with high need for work coordination. These teams work with interdependent tasks and in an environment with many procedures to follow (high process complexity). In the fourth group is team A, which has exceptionally high needs for work coordination, because this team works on tasks with very high task interdependence. In order to discuss these groups, we follow the same structure for all groups. We first report on the familiarity of the team with the process and the stability of processes and the quality of the process documentation in the project. Then we discuss process complexity in terms of two factors—task interdependency and process dependency, and finally, the characteristics of the team-external work coordination are considered. The summary of these results is found in Table 9.

Group 1 (Team G and Team H)

In this group, needs for work coordination are low. Both teams in this group are from Project B, where processes have been stable over time, and thus, the teams have high familiarity with the processes.

Both teams in the group are from Project B, which has been following a V-model development process for a very long time. Both teams have members that are experienced and familiar with processes. The process is well-documented as all teams must follow guidelines for safety-critical software systems development.

For both teams in this group, process complexity is moderate, as Team G and Team H from Project B are working on designated subsystems and thus deal with relatively independent tasks. However, process dependencies are high. Both teams have task-level dependencies because tasks

require mandatory agreements with safety engineers for reviews, guidelines, documents for safety-critical reasons, and system-level dependencies. This is because the teams work on a complex product that involves interactions with hardware.

Both teams have a relatively small number of team-external work coordination contacts, as Team H reported 8 team-external work coordination contacts (1.60 contacts per participant), and Team G reported 7 contacts (1.40 contacts per participant).

Group 2 (Team D, Team E, and Team F)

In this group, needs for work coordination are moderate. Teams work with relatively isolated tasks, and all teams are from an offshore site from Company A, where process stability has been moderate at the time of the investigation.

Teams in this group are relatively new. Therefore, process familiarity is moderate to low because their knowledge about the process and ways of working is also relatively low, especially in the case of team F. Furthermore, team E perceives that there is room for improvement in better understanding about processes. Process stability is moderate, as Company A has recently changed some of the processes. The teams have determined that the quality of process documentation is average and that this factor reduces the needs for work coordination.

Process complexity for all teams in this group is low. The relatively inexperienced offshore teams in Project A receive tasks with fewer task interdependencies. Process dependencies are relatively moderate as teams generally need to coordinate with technical experts (whom they need to involve in approving solutions) and testers. An individual from Team E indicated that there are requirements to handshake solutions with experts, such as system managers, design management teams, maintenance teams, function testers, system testers, and co-integration managers. Sometimes team F finds it difficult to agree on task solutions with technical experts, which leads to rework.

Team D reported 12 contacts (1.50 contacts per participant), and Team E reported 13 team-external work coordination contacts (2.60 contacts per participant), which in our sample is average. However, Team F reported only two team-external work coordination contacts (0.33 contacts per participant), which is low.

Group 3 (Team B, Team C, and Team I)

Teams in this group show high needs for work coordination. Although all teams in this group are senior teams in the company, they also work on tasks with a lot of interdependencies and must adhere to multiple and complex procedures.

Process familiarity for all teams in this group is moderate. However, the reasons are different. As previously mentioned, Project B follows the V-model, and processes have been stable, but Team I has been involved in the new development subproject with different ways of working. Team B and Team C are challenged by frequent process changes in their site, as onsite is used to introduce new processes.

The process complexity for all teams in this group is high. Team B and Team C from Project A work at an onsite office and receive diverse and interdependent tasks in different parts of the system. Team I has more team-external work coordination with others because Team I is responsible for the module that is interrelated with other subsystems. Working across subsystems also leads to more process dependencies, because it usually requires the team to anchor their solutions with experts outside the team, such as technical experts and architects in different subsystems, as Team I noted: *We feel that we communicate much* when discussing work coordination with others outside the team. The approach to work coordination in these teams is slightly different.

Team C and Team B indicated that they rely more on their experts in a team to address process dependencies, and Team C reported nine team-external work coordination contacts (1.80 contacts per participant). However, Teams B reported an above-average value of 14 for team-external work coordination contacts (2.80 contacts per participant). Team I indicated that it is actively coordinating with others but has reported 11 team-external work coordination contacts (2.75 contacts per participant).

Group 4 (Team A)

Team A has a very high needs for work coordination. This is a senior team in the company. Team A is differentiated from Group 3 teams in the same company in that this team receives tasks that have even more dependencies with other subsystems and layers.

Process familiarity for Team A is low, as they are challenged with recent changes in processes at their site, as the main office is used to introduce new processes in the project. For example, Team A is also involved in introducing many changes as a common object for pilot studies and trials: The team is the driver for continuous integration effort and the need for better tool support. Being a pilot team also means that process documentation may not be established yet.

Process complexity for Team A is very high because the members are working on tasks that have very high interdependency, as a team member noted: *As far as I know, it is only [Team A] and (another) that work across sub-systems in this way.* Team A also indicated that addressing process dependencies requires significant effort: *Gathering information and deciding what to do and so on. What needs to be done and how it should be done. It is hard to understand each time how to do it; you have to look at the process every time.*

Team A has reported a very large number of team-external work coordination contacts with 30 unique team-external work coordination contacts (5.00 contacts per participant), which also includes a lot of informal experts and former members from the project.

4.3 | Relationship between team-external coordination and teams' performance

To understand the relationship between satisfaction of the teams' coordination needs by team-external coordination and how it relates to teams' performance, we have mapped the previously discussed findings with team performance data (see Figure 6). In the figure on the y-axis, we show respective teams' needs: need for expertise coordination and need for work coordination. On the x-axis, we show respective reported team-external coordinating contacts for two categories: expertise coordination and work coordination. Size is reported as the number of contacts per team member to normalize differences in participated team members and thus a comparison between teams. When teams are positioned on the diagonal (that goes through average size of reported coordination contacts point), this means that the actual number of coordination contacts has the potential to satisfy the needs for expertise and work coordination (the higher the teams' needs, the larger the number of team-external coordination contacts). Performance is visualized using a heat map principle. Each colored point in the figure is one team, and the color indicates team performance from 1 (red) to 5 (green). We expect that satisfied teams' needs will result in a better performance by the team, while unsatisfied needs will result in underperformance by a team. Both needs—for expertise and work coordination—should be satisfied to produce better performance in a team. Our results from analysis of teams' needs for expertise coordination and team-external expertise coordination contacts show that two out of four high performing teams are on or above the diagonal (Team I and Team G). However, two teams (Team A and Team C) are below the diagonal, whereby even though they have reported a small number of team-external expertise coordination contacts, they are performing very well. All four average performing teams are on or slightly below the line. Team F has very low performance and is also below the line. Our results for teams' need for work coordination and team-external work coordination do not show clear individual patterns. Team F has reported a very small number of team-external work coordination contacts and is underperforming. Team A has reported an exceptionally large number of team-external work coordination contacts and is above the line. Team C, again, is below the line. Overall Team E, Team G, and Team I are above the line in the expertise coordination category and above or close to the line in work coordination category, indicating that these two teams might have fulfilled their needs. Team A is below the line in expertise coordination category but high above the line in the work coordination category. Team B is below the line in expertise coordination category but close to the line in work coordination category and has shown good performance. Team H very close to the line in both categories and has shown average performance. Three teams are below the line in both categories. Team D is slightly below in both coordination needs categories and shows average performance, while Team F is well below in both teams' coordination needs categories and shows weak performance. This indicates that their teams' coordination needs are not fulfilled. Although Team C is also below the line in both coordination needs categories, this team is assessed as performing very well.

5 | DISCUSSION

We conducted a multicase study in which the unit of analysis was a software development team in a large-scale distributed software development project. Our focus was on understanding coordination in large-scale projects and the impact of team-external coordination. We had access to two large-scale software development projects, which allowed us to study nine software development teams in total. We were able to gather data about team-external coordination and the content of coordination activities. We have described coordination activities in large-scale software development projects using the definition by Malone and Crowston—*coordination is managing dependencies between activities*²⁶—as our analytical

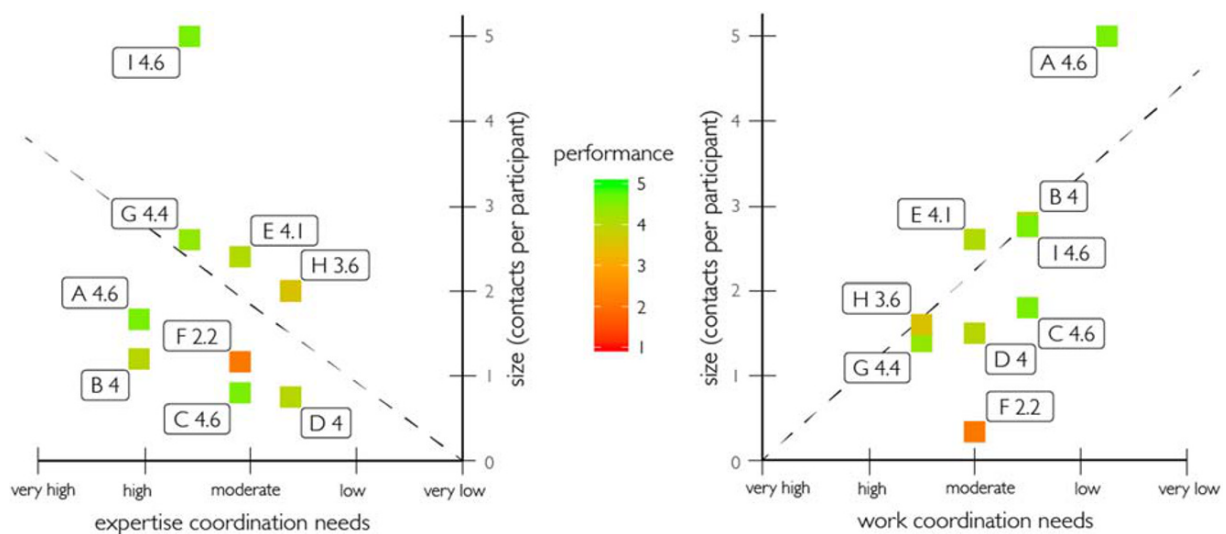


FIGURE 6 Team performance versus teams' coordination needs and team-external coordination

framework. We have shown that in the development of large-scale software systems, dependencies between different modules or services are managed by team members coordinating with other team members, external experts, and supporting roles. While individual skills and knowledge have been considered to be important for team performance,^{51,64} we argue that knowledge and skills are not enough to achieve performance in large-scale projects. In large-scale projects, dependencies between activities need to be managed effectively. Moreover, in large-scale projects, unplanned decentralized communication is essential for rapid responses when handling dependencies and therefore essential for team performance. We observed that communication in our large-scale study was mainly between teams, experts, and supporting roles, which we argue is a sign of decentralized communication. Decentralized communication is also an indicator of decentralized coordination and control.

We found several project-, team-, and task-related characteristics that impact teams' coordination needs, for example, complex tasks require the team to coordinate with many experts. Our study confirms the findings of Smite et al.¹⁷ that found that networking is essential for teams when solving complex, unfamiliar, and interdependent tasks. Our study extends Smite et al.¹⁷ findings by identifying different types of dependencies between activities and linking different project-, team-, and task-related characteristics to teams' coordination needs in two categories: (i) expertise coordination and (ii) work coordination. Expertise coordination is needed to address dependencies on one of the critical resources for knowledge teams - expertise.¹⁶ For example, the software development team needs to interact with experts in another team to solve a task. Work coordination is needed to address a variety of constraints in the workflow, for example, when a team needs to integrate their effort with others, and the output from another team is needed as input to do their tasks. We also demonstrate that team-external coordination is often characterized by a large number of external contacts. That is, the team needs to communicate directly with many other teams, experts, and support roles, which highlights the importance of team-external coordination and decentralized communication. Furthermore, we found that teams often have many team-external coordination contacts to satisfy teams' coordination needs. Additionally, software development teams have different team-external coordination patterns, which depend on teams' coordination needs, highlighting that it is essential to understand teams' coordination needs to understand communication and coordination in larger-scale projects.

5.1 | Characteristics of teams' coordination needs in large-scale software development

To answer our first research question (**What characterizes teams' coordination needs in large-scale software development?**), we collected data on team member contacts with other teams, experts, and supporting roles and classified reported team-external coordination connections. In focus groups, we also collected data about each team and their daily work, which allowed us to characterize teams' coordination needs.

Our analysis of team-external coordination patterns and analysis from focus groups suggests that development teams in large-scale projects often have many contacts for coordination. Furthermore, our results suggest that teams often engage in decentralized communication, which is essential for rapid responses when handling interdependencies. Our results suggest two explanations for the high number of contacts and the need for decentralized communication. The first explanation is that for teams that are working on large-scale systems, demands for expertise are often very high, as it is rarely realistic for one team to understand the entire large-scale system,^{35,36,53} and teams need to compensate for the knowledge gaps.^{3,16,45} We found that highly experienced teams tend to receive more complex and interdependent tasks and may be challenged by tasks in unfamiliar product areas, increasing the need for expertise coordination. Therefore, teams that receive more complex tasks need to interact with many supporting roles, for example, system architects, system managers, and area experts. While new teams had a much smaller network than experienced teams, we still found that they had a high need for expertise coordination because they lacked the knowledge. The second explanation for the high number of external contacts and the decentralized communication is that teams in large-scale projects work in a multiteam environment and depend on supporting roles for certain functions (for example, requirements, testing, and deployment).^{4,9,10,61} Our findings confirm previous research on coordination^{4,12,18} that has demonstrated that inter-team coordination by mutual adjustment plays an essential role in large-scale software development. Furthermore, even when projects use a modularized system architecture, coordination is still necessary to solve work dependencies among teams^{9,39}.

Our sample is too small for robust findings; however, an interesting observation was made concerning the differences between teams' coordination needs across the investigated projects. We found that component teams and feature teams possessed different needs and exhibited different team-external coordination patterns. Component teams show lower work coordination than the feature teams that have more coupled work, although these needs still exist and are not negligible. Our results suggest the explanation is lower task interdependence. We also determined that experienced component teams have higher expertise coordination than experienced feature teams because they are required to have basic knowledge about a number of unfamiliar components (Table 8).

Results of semantic analysis of the project-, team-, and task-related characteristics and the analysis of teams' coordination needs show that the actual teams' coordination needs might vary greatly between projects and even within the same project. The teams studied related the expertise and work coordination needs directly to various task and project characteristics, namely, task interdependence,^{67,68} process complexity,⁶⁷ parallel and interdependent work with other teams,¹ and product and process coverage with documentation.^{17,66}

We found that key factors that determine teams' needs for expertise coordination are task familiarity and complexity. Our results confirm that team members' experience in software development and team members' experience in the project will impact teams' needs for expertise

coordination.¹⁶ Teams' joint work experience will contribute to the effective use of the team knowledge pool,^{40,65} reducing the need to seek expertise elsewhere. Furthermore, our results show that having an expert member on the team reduces the need for external knowledge pull. However, we also learned that if these experts only work part-time on the team, their limited availability does not allow the team's needs for expertise to be fully satisfied. Finally, we determined that teams with multiple new members experience difficulty in satisfying teams' coordination needs because of the limited familiarity with project members outside of the team, and the low awareness of the knowledge level of members, and an inability to identify the experts. The challenge of locating expertise and awareness has also been noted in existing research on large-scale software development.¹²

We found that key factors that determine teams' work coordination needs are process familiarity and complexity. We found that new teams can have low work coordination needs because they tend to work on simpler and independent tasks,⁷⁶ as long as processes are stable in the project.¹⁷ Teams' work coordination needs will decrease when teams become more familiar with processes. However, we found that more experienced teams also tend to receive increasingly more complex and interdependent tasks, requiring them to solve more work dependencies, such as agreement on design solutions, doing code reviews, handshaking on solutions with area experts, and addressing task dependencies with other teams. Task interdependencies and, therefore, work coordination needs also increase as the teams work with tasks that have dependencies with other subsystems and layers.⁵ Good process documentation was mentioned as one way to decrease work coordination needs for teams working in unfamiliar areas.⁶⁶

In summary, our results confirm findings of previous research that demonstrated that expertise coordination^{10,16} and work coordination^{9,10} are important parts of overall coordination efforts by teams in software development projects. Therefore, it is fair to state that maintaining team-external coordination contacts is important throughout the entire project life cycle. On the basis of the reported team-external coordination patterns, we observed that experienced teams also have more team-external coordination contacts on an individual team member basis, which must be a result of continuous contact growth over time. Based on the way they describe the use of team-external coordination at work, it seems that experienced teams are comfortable relying on their team-external coordination contacts and decentralized communication, and they have the ability to find the right experts when needed, instead of investing time in the process of building the knowledge internally or finding and studying documentation.

5.2 | Team performance in large-scale software development

To answer our second research question (**What is the relationship between team-external coordination and teams' performance in large-scale software development?**), we also collected data on team performance. Our results from performance data (illustrated in Figure 7 and summarized in Figure 8) suggest that fulfilling teams' coordination needs by team-external coordination in both categories, (1) expertise coordination and 2) work coordination, seems to be important for improved team performance in large-scale software development. The results contribute to existing research related to the importance of expertise^{16,53} and work coordination.^{9,12,35} Evidently, we have outliers two teams with high performance and unsatisfied needs (from the perspective of our definition of the needs) in one or both categories (Teams C and G) and one team with average performance and satisfied needs (Team H). One possible explanation could be that their needs were overestimated and that, for example, the influence of formal experts in the team is greater than we assumed. In the case of Team C, there are two expert members (one who currently is a formal expert and one who was previously a formal expert). Another challenge is to understand the influence of expertise coordination needs versus work coordination needs because coordination in our case is a compound variable. We observed that unsatisfied work coordination needs do not appear to influence performance more than unsatisfied expertise coordination needs (based on Team G). We observed that Team A oversatisfied their needs of work coordination but has undersatisfied needs in expertise coordination and performs well. Moreover, it is evident that teams that fulfill both needs perform well (Teams E and I), while teams that do not fulfill their needs in both categories perform worse (Teams D and F). In summary, our results provide strong indications that team-external coordination matters for team performance. Unfortunately, given that our sample of teams is relatively small, we do not have a sufficient number in each of the groups (see Figure 8 for firm conclusions). As such, the direction of future research is to gather more evidence in a study with more teams with different profiles and different team-external coordination patterns (symmetrical and asymmetrical satisfaction of expertise coordination and work coordination needs).

5.3 | Limitations

There are some evident limitations in the reliability and generalizability of our findings. The first limitation is that much of the data collection and analysis was based on semi-structured interviews and focus groups. The consequence of this limitation is that the results are influenced by our interpretation of the phenomena observed and investigated. The use of multiple data sources (interviews, focus groups, and survey) made it possible to confirm evidence for the team coordination needs. The study included a survey, as well as talking to and interviewing team members and

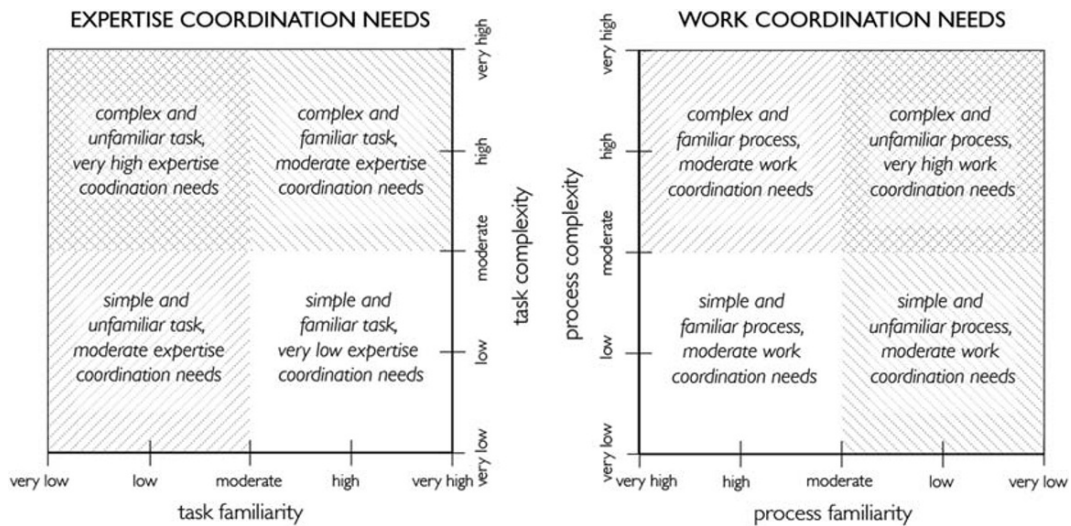


FIGURE 7 Characteristics that impact teams' coordination needs

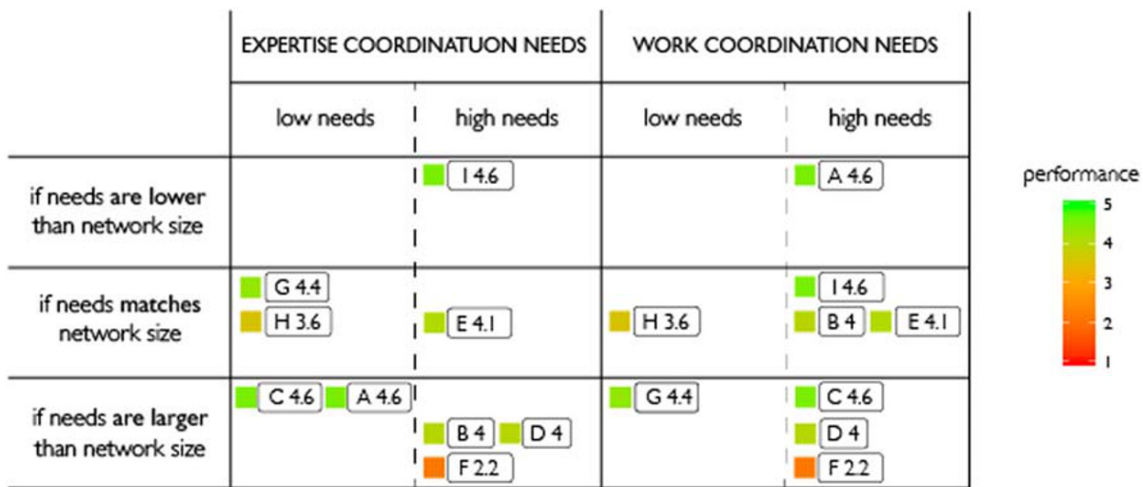


FIGURE 8 Relationship between the teams' needs, satisfaction of these needs by team-external coordination, and teams' performance

managers in both companies and at all sites of the teams investigated, which made it possible to investigate the phenomena from different viewpoints, thus reducing this limitation. There is a risk that our findings can also be explained by factors that evaded our attention. One reason is that we did not collect data from observations on actual team-external coordination but relied on self-reported data from the participants; therefore, we probably missed some team-external coordination patterns. In other words, we investigated the perception of the coordination and did not attempt to calculate or determine an objective measurement of the social network. Consequently, expertise and information sharing could have been exaggerated or underestimated by the respondents. Additionally, we collected team-external coordination data only once at a certain time point of the study. It is evident that coordination patterns can change over time based on changes in needs for expertise coordination and work coordination, as well as changes based on processes in the projects and personnel changes. However, giving feedback to the observed teams and managers and discussing our interpretation of the team-external coordination patterns helped validate our conclusions. Another consequence of this limitation is related to performance measurements. We used team performance evaluations, which are inherently subjective. Our use of this type of performance measurement is grounded in the fact that there are no available objective measures that can be used for measurements across multiple companies and facilitate comparisons in the same manner. However, discussing our interpretation with managers and the use of internal validity measurements helped improve the reliability of performance measurements. The second limitation is the use of magnitude coding for the analysis of the teams' needs from qualitative data obtained in semi-structured interviews and focus groups. Magnitude coding relies on researcher judgment to convert qualitative data into quantitative data. We mitigated this by triangulation. As the first step, the first and the second author coded qualitative data independently, and as the second step, both authors discussed any disagreements to decide on the final code. Additionally, the provisional coding analysis of the questionnaire might be sensitive to the classification of team-external coordination

connections and further use in analysis of coordination and patterns. To reduce this limitation, the first and the second author coded qualitative data independently, and as the second step, both authors discussed any disagreements to decide on the final classification of each coordination connection. When converting quantitative scales into qualitative, special attention was given to discussing the convergence of the data toward a single scale and the necessity of taking the case-specific context into account. For example, what is very large in one context might be small in another context. In addition, we were limited to the data points available, and our magnitude and provisional coding might have been different if we had teams in our sample that exhibited even more team-external coordination. Further replications of our study in other contexts and with more teams could help in calibrating our magnitude scale and reducing this threat to conclusion validity; however, we believe that triangulation and discussion of the results with the case representatives helped to sufficiently mitigate this threat.

The relationship model of teams' coordination needs and their influence by team-external coordination and team performance also have limitations. Our model is based on a complex and real work environment of large-scale distributed development projects, which limits our ability to control variables that may influence team performance in such settings. This reduced the validity of our findings, for example, false attribution of team-external coordination behavior to teams' coordination needs or underestimating or overestimating teams' coordination needs and fully capturing all factors that impact teams' coordination needs. To assure Content Validity, we have discussed preliminary findings and conclusions of this study with the representatives from both companies involved. We received feedback that showed that conclusions were valid at the time of the study. We cannot guarantee that replicating the study in the same or similar contexts would lead to the conclusions. However, given the state of practice and tool advancement at the time of data collection and today, there is no reason to believe that the link between the satisfaction of the team coordination needs and team performance is not true today. Moreover, a recent application of the survey instrument for creating social networks in a modern product environment utilizing microservice architecture showed that the content and amount of coordination are comparable and that teams that were seen more successful had larger networks, while teams that were challenged had fewer contacts.⁹⁰

Further studies with additional data samples are needed to understand the full complexity of what factors impact team-external coordination in a large-scale distributed environment. Further studies that sample more teams with potentially different needs and team-external coordination patterns are needed to further understand the interplay between team-external coordination and team performance. Finally, the generalizability of the conclusions is impacted by the sampling strategy and the cases included in our investigation. We were only able to access two large-scale distributed software development projects at two companies. This limited our ability to use a more rigorous sampling technique in terms of team-external coordination research and to limit control variables of the teams' coordination needs. However, this allowed us access to the otherwise unavailable context, the large-scale distributed development for this type of study. We combined several instruments to improve the reliability of our conclusions (e.g., multiple data sources and cross-case triangulation). Gathering empirical data within similar conditions is required to generalize our findings.

6 | CONCLUSIONS

In this paper, we share the findings from an exploratory study of teams' coordination needs, especially when challenged with unfamiliar or complex tasks. Teams working in the same project often will have different needs depending on a variety of factors and thus should be treated differently. This means, for example, that mandatory coordination or communication fora and rituals might not be equally beneficial for everyone. Organizations should instead focus on enabling the teams to source expertise and address work dependencies with the right contacts when needed. Furthermore, to improve the support of teams, organizations should improve the awareness and availability of experts and support roles for teams. Organizations also should support the necessary infrastructure for local and cross-site expertise and work coordination. Finally, our study suggests that the satisfaction of teams' coordination needs is important for achieving desired performance. We suggest that future research should focus on extending the evidence of teams' coordination needs and team-external coordination patterns and their effect on team performance by increasing the number and diversity of cases, teams, projects, and task characteristics. We acknowledge that cultivating team-external coordination can require time and effort that could also be used for development work. Thus, future research should also explore cases in which too extensive team-external coordination can hamper team performance.

ACKNOWLEDGMENT

This research was funded by the Swedish Knowledge Foundation under the KK-Hög grant 2009/0249 within the TEDD research project and under the KK-Hög grant 2016/0191 within the GoLD research project and the A-team project supported by the Research council of Norway through grant 267704 and the companies Kantega, Sbanken, Storebrand, and Knowit.

ORCID

Aivars Sablis  <https://orcid.org/0000-0003-2972-139X>

REFERENCES

1. Perry DE, Siy HP, Votta LG. Parallel changes in large-scale software development: an observational case study. *ACM Trans Softw Eng Methodol (TOSEM)*. 2001;10(3):308-337.
2. Petersen K, Roos P, Nyström S, Runeson P. Early identification of bottlenecks in very large scale system of systems software development. *J software Evol Process*. 2014;26(12):1150-1171.
3. Curtis B, Krasner H, Iscoe N. A field study of the software design process for large systems. *Commun ACM*. 1988;31(11):1268-1287.
4. Paasivaara M, Lassenius C, Heikkilä VT. Inter-team coordination in large-scale globally distributed scrum: do scrum-of-scrums really work. In: Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement ACM; 2012:235-238.
5. Herbsleb JD, Grinter RE. Architectures, coordination, and distance: Conway's law and beyond. *IEEE software*. 1999;16(5):63-70.
6. Herbsleb JD, Mockus A. An empirical study of speed and communication in globally distributed software development. *IEEE Trans softw eng*. 2003;29(6):481-494.
7. Dingsøy T, Fægri TE, Itkonen J. What is large in large-scale? A taxonomy of scale for agile software development. Springer; 2014:273-276.
8. Britto R, Cruzes DS, Smite D, Sablis A. Onboarding software developers and teams in three globally distributed legacy projects: a multi-case study. *J Softw Evol Process*. 2018;30(4):e1921.
9. Espinosa JA, Slaughter SA, Kraut RE, Herbsleb JD. Team knowledge and coordination in geographically distributed software development. *J manag infor syst*. 2007;24(1):135-169.
10. Strode DE. A dependency taxonomy for agile software development projects. *Infor Syst Frontiers*. 2016;18(1):23-46.
11. Petersen K, Wohlin C. The effect of moving from a plan-driven to an incremental software development approach with agile practices. *Empir Softw Eng*. 2010;15(6):654-693.
12. Dingsøy T, Bjørnson FO, Moe NB, Rolland K, Seim EA. Rethinking coordination in large-scale software development ACM; 2018:91-92.
13. Dingsøy T, Conradi R. A survey of case studies of the use of knowledge management in software engineering. *Int j softw eng knowledge eng*. 2002;12(04):391-414.
14. Kankanhalli A, Tan BC, Wei KK. Contributing knowledge to electronic knowledge repositories: an empirical investigation. *MIS quarterly*. 2005;29(1).
15. Bjørnson FO, Dingsøy T. Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used. *Inf Softw Technol*. 2008;50(11):1055-1068.
16. Faraj S, Sproull L. Coordinating expertise in software development teams. *Manag Sci*. 2000;46(12):1554-1568.
17. Smite D, Moe NB, Sablis A, Wohlin C. Software teams and their knowledge networks in large-scale software development. *Inf Softw Technol*. 2017;86:71-86.
18. Moe NB, Dingsøy T. Emerging research themes and updated research agenda for large-scale agile development: a summary of the 5th international workshop at XP2017 ACM; 2017:14.
19. Dingsøy T, Moe NB. Towards principles of large-scale agile development. In: International Conference on Agile Software Development Springer; 2014:1-8.
20. Petersen K, Wohlin C, Baca D. The waterfall model in large-scale development. Springer; 2009:386-400.
21. Pernstål J, Gorschek T, Feldt R, Florén D. Requirements communication and balancing in large-scale software-intensive product development. *Infor Softw Technol*. 2015;67:44-64.
22. Sarkar S, Kak AC, Rama GM. Metrics for measuring the quality of modularization of large-scale object-oriented software. *IEEE Trans Softw Eng*. 2008;34(5):700-720.
23. Ackerman MS, Halverson C. Considering an organization's memory ACM; 1998:39-48. now 23 now 23
24. Ågerfalk J, Fitzgerald B. OP. Flexible and distributed software processes: old petunias in new bowls; Citeseer.:2006.
25. Malone TW, Crowston K. The interdisciplinary study of coordination. *ACM Comput Surv (CSUR)*. 1994;26(1):87-119.
26. Yuan M, Zhang X, Chen Z, Vogel DR, Chu X. Antecedents of coordination effectiveness of software developer dyads from interacting teams: an empirical investigation. *IEEE Trans Eng Manag*. 2009;56(3):494-507.
27. Dietrich P, Kujala J, Arto K. Inter-team coordination patterns and outcomes in multi-team projects. *Proj Manag J*. 2013;44(6):6-19.
28. Nguyen-Duc A, Cruzes DS, Conradi R. The impact of global dispersion on coordination, team performance and software quality—A systematic literature review. *Inf Softw Technol*. 2015;57:277-294.
29. Blackburn JD, Scudder GD, Van Wassenhove LN. Improving speed and productivity of software development: a global survey of software developers. *IEEE trans softw eng*. 1996;22(12):875-885.
30. Gregor S. The nature of theory in information systems. *MIS quarterly*. 2006:611-642.
31. Strode DE, Huff SL, Hope B, Link S. Coordination in co-located agile software development projects. *J Syst Softw*. 2012;85(6):1222-1238.
32. Crowston K, Rubleske J, Howison J. Coordination theory: a ten-year retrospective. *Human-computer interaction and management information systems: Foundations*; 2006:134-152.
33. Malone TW, Crowston K. What is coordination theory and how can it help design cooperative work systems. In: Proceedings of the 1990 ACM conference on Computer-supported cooperative work; 1990:357-370.
34. Kraut RE, Streeter LA. Coordination in software development. *Commun ACM*. 1995;38(3):69-82.
35. Bjarnason E, Wnuk K, Regnell B. Requirements are slipping through the gaps—a case study on causes & effects of communication gaps in large-scale software development IEEE; 2011:37-46.
36. Ulziit B, Warraich ZA, Gencel C, Petersen K. A conceptual framework of challenges and solutions for managing global software maintenance. *J Softw Evol Process*. 2015;27(10):763-792.
37. Paetsch F, Eberlein A, Maurer F. Requirements engineering and agile software development IEEE; 2003:308-313.
38. Cataldo M, Herbsleb JD. Coordination breakdowns and their impact on development productivity and software failures. *IEEE Trans SoftwEng*. 2013;39(3):343-360.
39. Moe NB, Fægri TE, Cruzes DS, Faugstad JE. Enabling knowledge sharing in agile virtual teamsIEEE; 2016:29-33.
40. Herbsleb JD, Mockus A, Finholt TA, Grinter RE. An empirical study of global software development: distance and speed. In: Proceedings of the 23rd International Conference on Software Engineering. ICSE 2001 IEEE; 2001:81-90.

41. Deshpande S, Beecham S, Richardson I. Global software development coordination strategies-a vendor perspective. In: International Workshop on Global Sourcing of Information Technology and Business Processes Springer; 2011:153-174.
42. Begel A, Nagappan N, Poile C, Layman L. Coordination in large-scale software teams. In: 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering IEEE; 2009:1-7.
43. Nyrud H, Stray V. Inter-team coordination mechanisms in large-scale agile, Vol. 16 ACM; 2017.
44. Paasivaara M, Lassenius C. Communities of practice in a large distributed agile software development organization-Case Ericsson. *Inf Softw Technol.* 2014;56(12):1556-1577.
45. Dingsøy T, Rolland K, Moe NB. Coordination in multi-team programmes. An investigation of the group mode in large-scale agile software development. *Procedia Comput Sci.* 2017;121:123-128.
46. Moe NB, Dingsøy T, Rolland K. To schedule or not to schedule? An investigation of meetings as an inter-team coordination mechanism in large-scale agile software development; 2018.
47. Perry DE, Staudenmayer NA, Votta LG. People, organizations, and process improvement. *IEEE Softw.* 1994;11(4):36-45.
48. Crowston K. A coordination theory approach to organizational process design. *Org Sci.* 1997;8(2):157-175.
49. Amrit C. *Improving coordination in software development through social and technical network analysis*: University of Twente; 2008.
50. Johnson M, Senge M. Learning to be a programmer in a complex organization. A case study on practice-based learning during the onboarding process at Google. *J Work Learn.* 2010;22(3):180-194.
51. Zhou M, Mockus A, Weiss D. Learning in offshored and legacy software projects: How product structure shapes organization; 2009.
52. Rus I, Lindvall M. Knowledge management in software engineering. *IEEE softw.* 2002;19(3):26.
53. Sengupta B, Chandra S, Sinha V. A research agenda for distributed software development. In: Proceedings of the 28th international conference on Software engineering; 2006:731-740.
54. Earl M. Knowledge management strategies: Toward a taxonomy. *J manag infor syst.* 2001;18(1):215-233.
55. Dingsøy T, Røyrvik E. An empirical study of an informal knowledge repository in a medium-sized software consulting company IEEE Computer Society; 2003:84-92.
56. Dingsøy T, Djarraya HK, Røyrvik E. Practical knowledge management tool use in a software consulting company. *Commun ACM.* 2005;48(12):96-100.
57. Nahapiet J, Ghoshal S. Social capital, intellectual capital, and the organizational advantage. *Acad manag rev.* 1998;23(2):242-266.
58. Akgün AE, Byrne J, Keskin H, Lynn GS, Imamoglu SZ. Knowledge networks in new product development projects: a transactive memory perspective. *Infor manag.* 2005;42(8):1105-1120.
59. Mathiassen L, Vogelsang L. The role of networks and networking in bringing software methods to practice. IEEE; 2005:256b-256b.
60. Hannay JE, Benestad HC. Perceived productivity threats in large agile development projects ACM; 2010:15.
61. Tiwana A, Mclean ER. Expertise integration and creativity in information systems development. *J Manag Inf Syst.* 2005;22(1):13-43.
62. Bauer TN, Erdogan B. Organizational socialization: The effective onboarding of new employees.; 2011.
63. Huckman RS, Staats BR. Team familiarity, role experience, and performance: evidence from Indian software services. *Manag Sci.* 2009;55(1):85-100.
64. Lewis K. Measuring transactive memory systems in the field: scale development and validation. *J Appl Psychol.* 2003;88(4):587-604.
65. Ramesh B, Cao L, Mohan K, Xu P. Can distributed software development be agile. *Commun ACM.* 2006;49(10):41-46.
66. Ovaska P, Rossi M, Marttiin P. Architecture as a coordination tool in multi-site software development. *Softw Process: Improve Prac.* 2003;8(4):233-247.
67. Ko AJ, Myers BA, Coblenz MJ, Aung HH. An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks. *IEEE Trans softw eng.* 2006;32(12):971-987.
68. Andres HP, Zmud RW. A contingency approach to software project coordination. *J Manag Infor Syst.* 2002;18(3):41-70.
69. Dingsøy T, Moe NB, Seim EA. Coordinating knowledge work in multiteam programs: findings from a large-scale agile development program. *Proj Manag J.* 2018;49(6):64-77.
70. Herbsleb JD. Global software engineering: the future of socio-technical coordination. In: Future of Software Engineering (FOSE'07) IEEE; 2007: 188-198.
71. Martini A, Pareto L, Bosch J. A multiple case study on the inter-group interaction speed in large, embedded software companies employing agile. *J Softw Evol Process.* 2016;28(1):4-26.
72. Van De Ven AH, Delbecq AL, Koenig R Jr. Determinants of coordination modes within organizations. *Am sociolo rev.* 1976:322-338.
73. collab=Lunenburg FC. Organizational structure: Mintzberg's framework. *Int j scholar acad intell diver.* 2012;14(1):1-8.
74. Dingsøy T, Moe NB, Fægri TE, Seim EA. Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. *Empir Softw Eng.* 2018;23(1):490-520.
75. Espinosa JA, Slaughter SA, Kraut RE, Herbsleb JD. Familiarity, complexity, and team performance in geographically distributed software development. *Org sci.* 2007;18(4):613-630.
76. Moe NB, Smite D, Sblis A, Börjesson AL, Andréasson P. Networking in a large-scale distributed agile project, Vol. 12 ACM; 2014.
77. Runeson P, Host M, Rainer A, Regnell B. *Case Study Research in Software Engineering: Guidelines and examples*: John Wiley & Sons; 2012.
78. Stewart DW, Shandasani PN. Focus groups: Theory and Practice. 20 Sage publications; 2014.
79. Sjoberg DI, Dyba T, Jorgensen M. The future of empirical methods in software engineering research IEEE; 2007:358-378.
80. Marsden PV. *Survey Methods for Network Data*, The SAGE handbook of social network analysis, vol. 25; 2011:370-388.
81. Manteli C, Van Den Hooff B, Van Vliet H. The effect of governance on global software development: An empirical research in transactive memory systems. *Inf Softw Technol.* 2014;56(10):1309-1321.
82. Heath S, Fuller A, Johnston B. Chasing shadows: defining network boundaries in qualitative social network analysis. *Qual Res.* 2009;9(5):645-661.
83. Guinan PJ, Coopridge JG, Faraj S. Enabling software development team performance during requirements definition. *A behavioral versus technical approach. Information systems research.* 1998;9(2):101-125.
84. Saldaña J. *The Coding Manual for Qualitative Researchers* Sage; 2015.
85. DeCuir-Gunby JT, Marshall PL, McCulloch AW. Developing, using a codebook for the analysis of interview data. An example from a professional development research project. *Field methods.* 2011;23(2):136-155.
86. Fruchterman TM, Reingold EM. Graph drawing by force-directed placement. *Softw Prac exp.* 1991;21(11):1129-1164.

87. Kline RB. Principles and practice of structural equation modeling Guilford publications; 2015.
88. DeVellis RF. Scale development: Theory and applications. 26 Sage publications; 2016.
89. Šmite D, Moe NB, Floryan M, Gonzalez-Huerta J, Šablīs A. Squad Autonomy at Spotify at Scale. submitted to IEEE Software, Special issue on Behavioral Software Engineering; 2020.

How to cite this article: Sablis A, Smite D, Moe N. Team-external coordination in large-scale software development projects. *J Softw Evol Proc*. 2020;e2297. <https://doi.org/10.1002/smr.2297>