Coordination in Large-Scale Agile Software Development

Marthe Berntzen(✉)

Department of Informatics, University of Oslo, Gaustadalléen 23B,
0730 Oslo, Norway
marthenb@ifi.uio.no

Abstract. Agile practices are popular within software development. But when applied to large projects with many teams, coordination challenges arise. The projects working title "*Coordination in large-scale agile software development: An investigation of coordination mechanisms, communication, roles, autonomy and interdependencies*" summarizes the main topics of investigation. While all theoretical and analytical approaches to the data material is not yet determined, I have already started fieldwork in one company which will serve as a main longitudinal case, with more to follow as the project proceeds. Initial fieldwork has revealed that there are differences in how agile teams coordinate their work across teams. I will continue to explore these differences. End goals of the project include to identify success criteria for coordination in large-scale agile software development projects.

Keywords: Large-scale agile · Software development · Coordination

1 Introduction

The main objective of this PhD project is to contribute to successful software development projects in the digital age, through contributing to a better understanding of coordination in large-scale agile software development projects. There is a recognized need for more research on how to adjust agile practices to large-scale contexts [1–4]. In particular, there is a need for more knowledge on coordination in autonomous agile teams in large-scale settings [3, 5, 6]. Here, I have the opportunity to study large-scale agile projects in a Scandinavian context, where companies seek inspiration from companies like Spotify and Ericsson.

This paper is organized as follows: Sect. 2 outlines some of the relevant background and related work on large-scale agile software development as well as some theoretical approaches to coordination. In Sects. 3 and 4 I present my preliminary research objectives and research design, while Sect. 5 outlines the next planned steps for the research project.

R. Hoda (Ed.): XP 2019 Workshops, LNBIP 364, pp. 123–133, 2019.
https://doi.org/10.1007/978-3-030-30126-2_15

2 Background and Relevant Work

2.1 Large-Scale Agile Software Development

Agile teams are autonomous and cross-functional in nature, where team members are assumed to make their own decisions and utilize their competence across different organizational functions and roles. This is thought to contribute to a flatter organizational structure with increased empowerment and participation, assumed to contribute to more efficient decision-making [6–9].

Despite that agile methods originally were intended for smaller team projects [10], and primarily has been successful in small teams [3], the practice of using agile principles and techniques has spread to include large-scale projects and organizations as a whole [9, 11]. At the same time, the research community on agile software development called out the need for a unified framework for understanding large-scale agile software development [12]. As a response to this, a taxonomy of scale for agile software projects was developed, where small-scale agile software development includes one team only, large-scale from 2–9 teams, and very large-scale from 10 teams and up [12].

When scaling up agile, several challenges arise, such as coordination between teams, stakeholder management and keeping to the agile principles [1, 3, 5]. One challenge with applying agile to large-scale is that there is a lack of a common, agreed upon understanding of agile working methods [13]. Rather, agile can be understood as a set of values, principles and practices, that may be implemented in more or less successful ways. As such, there may be great differences in how large-scale agile is implemented [14], and finding consistent results from large-scale agile may be difficult [13]. Initial fieldwork supports these observations. In Berntzen et al. [15] we discuss how differences in Product Owner coordination may be related to that teams in the large-scale case program under study may freely choose among agile methods, in other words, they do not work consistently with one agile approach.

Another challenge is related to how large-scale frameworks, such as the Spotify model, Large-Scale Scrum and the Scaled Agile Framework may affect large-scale coordination. Such frameworks are gaining in popularity, but there is still a need for more academic research on such practices, as there is little research supporting that agile principles can be directly applied to all organizational processes without adjustment or tailoring [1, 14, 16].

Among the many challenges inherent in the successful implementation of large-scale agile coordination appears to be a key issue. Dikert et al. [1] identify inter-team coordination as one of the major challenges in need of more research. Coordination, often defined as the managing of interdependencies [17] is recognized as important across literatures on software engineering, information systems, organization and management [26], and theories on coordination has been developed [4, 17]. While researchers have started exploring coordination in large-scale agile [1–3, 5, 16, 18, 19], there are still many open questions in need for further investigations.

2.2 Coordination Theories

Coordination Theory

Malone and Crowston [17] developed an interdisciplinary, broad-based theory of coordination, known today as Coordination Theory (CT). In their seminal paper, Malone and Crowston [17, p. 4] defined coordination as a process of "managing dependencies between activities". CT is based on ideas from organization theory, management, economics and computer science [4]. The basic tenet of CT is that complex organizational systems are made up of dependencies (such as shared resources, task interdependencies, simultaneity constraints and relationships with clients, each with different sub-dependencies), which constrain situational action, and thus must be coordinated. Coordination then, is made up by various coordination processes and mechanisms which each address one or more dependencies in a situation [17]. What these processes and mechanisms are and how they work vary with the context. In the context of large-scale agile software development, they can include for instance scheduled and unscheduled meetings, artefacts and physical settings [4, 20]. These mechanisms may facilitate action constrained by the dependencies, however, the in the large-scale setting, perhaps the mechanisms themselves may also both enable and constrain coordinated action?

CT has contributed with a much-cited definition of coordination, a modelling framework for analyzing coordination in complex processes and providing a beginning of a typology of dependencies and coordination mechanisms [21]. However, it does not provide any propositions or testable hypotheses [17, 21]. In a ten-year retrospective of CT research, future research to develop testable hypotheses from CT is encouraged, for instance about the generality of coordination mechanisms and more structured approaches to evaluate alternate coordination processes [21].

Despite the limitations of the theory in terms of lack of causal explanations and testable hypotheses, CT has proved a useful theoretical framework for the study of coordination. In the IS field, CT has been used in particular in software engineering and systems design, where researchers have noted the importance of coordination challenges and the potential for computer systems to help groups and teams collaborate better [21]. In the context of agile software development, CT has been applied by Strode and colleagues [4], who used the theory as basis for their own development of a theory of coordination in agile development.

The Theory of Coordination in Agile Development

To take advance theory and research on coordination in agile SD further, Strode and colleagues [4] build on Coordination theory but extended with a theoretical model and a total of eight testable propositions. In particular, this theory proposes that effective coordination in agile settings are comprised of coordination strategies contributing to coordination effectiveness. Coordination strategies are defined as a group of coordination mechanisms that manage dependencies in a situation. They consist of three components; synchronization, structure and boundary spanning activities and artefacts that contribute to overall coordination effectiveness [4].

Coordination effectiveness, in turn, consists of explicit and implicit effectiveness. Explicit coordination effectiveness emphasizes the physical objects (both persons and artefacts) involved in the project. For *explicit coordination effectiveness* to occur, the required object needs to be in the right place, at the right time and in the right state so that is "ready for use" as perceived by each individual involved in the project [4, 17]. Having the right tools in place to conduct a video meeting or having available developers to take on new tasks as they flow from a different team can be examples of this type of explicit coordination effectiveness. *Implicit coordination effectiveness* on the other hand, relates to coordination that occurs within work groups without explicit passing of messages. The authors further posit that implicit coordination consists of five components; "knowing why", "knowing what is going on and when", "knowing what to do and when", "knowing who is doing what" and "knowing who knows what". In other words, implicit coordination requires a high degree of shared goals and understanding both of one's own and others knowledge [4]. In relation to agile development, where the team is central [22], implicit coordination in terms of shared knowledge indeed appears important to overall project effectiveness.

Importantly, in this theory, these are considered outcomes resulting from the coordination strategy. The theoretical model proposes that there is a causal relationship between an agile coordination strategy and project coordination effectiveness; if the strategies are well implemented, coordination is more effective. This in turn, is proposed to contribute to the agile software development project success [4]. In addition, they propose that project complexity, uncertainty and organization structure may affect the coordination strategies, but they did not test this while developing the theory.

Despite its clear relevant to the study of coordination in agile development, this theory is difficult to readily apply it my PhD project because it considers *intra-team* coordination and does not consider the multiple team aspect and *inter-team* coordination, which may introduce important constraints to effective coordination. In order to apply their theoretical model to large-scale agile development, it could be necessary to expand the model to include elements such as for instance team size, number of teams, number of functional elements involved in the project as well as differences in team autonomy in their usage of agile methods and choice of technologies across teams. Accordingly, one route may be to further develop the theory to account for scale. Another route is to look further into theories that may take into account the multiple team aspect, and the various differences these entail, through focusing on the coordination process itself through a relational lens.

Relational Coordination Theory

Relational Coordination Theory (RCT) [23] represents a third theoretical perspective on coordination. RCT originates in the organization studies field from research conducted in the airline industry in the 1990s [23], where Gittell observed substantial difference between companies in the extent to which the employees shared collective goals and knowledge towards the overall work process and outcome. Today, RCT is an established and empirically validated theory, and has been studied in various (non-agile) large-scale settings, most notably in the airplane, health and education industries [24][1]. RCT has recently been picked up by Information Systems researchers [25–27], however, it appears it has not yet been applied in large-scale agile development.

Relational coordination is defined as "a mutually reinforcing process of interaction between communication and relationships carried out for the purpose of task integration" [28]. These relationships can be between individuals, roles or even departments and organizations. According to RCT, relationships provide the necessary bandwidth for coordinating work in settings with that are highly interdependent, uncertain and time-constrained. Effective coordination in these settings is carried out through relationships of shared goals, shared knowledge and mutual respect. These, in turn, are theorized to be mutually reinforced by high-quality communication (that is, frequent, timely, accurate and problem-solving communication). It is interesting to note that these assumptions bears resemblance to Strode et al.'s implicit coordination effectiveness [4] described in the above section. The resulting positive relational context enables a well-coordinated process with less wasted effort [23]. Finally, an assumption of RCT is that relational coordination has is stronger in more horizontally designed organizational structures [29]. Because large-scale agile software development processes are also typically characterized by high levels of interdependence, uncertainty and time pressure, in combination with other coordination theories, I believe RCT is an interesting lens for studying coordination in large-scale agile development.

Further Theoretical Considerations

Although the above presented theories all can contribute to the understanding of coordination processes in large-scale agile, it is still necessary to focus not only on the social and human aspects of coordination, but also the role of the product under development and the technologies being used during the development.

All three coordination theories offer some concepts that address coordination in large-scale agile development, however the role of large-scale itself, as well as the potential implications of both the technology being used for coordination, and the technology being developed is perhaps not fully addressed. In order to fully accommodate these theories to be relevant for large-scale agile development, and to make valuable theoretical contributions to IS and SE fields, it may be relevant to draw on other theories and concepts. As one overarching project goal is to address how coordination mechanisms are used in and across teams, and as initial fieldwork has indicated that teams in large-scale agile projects coordinate differently [15], it is important to address how different coordination mechanisms may be used in different ways. To this end, I believe that other theories and concepts from the IS field, such as sociotechnical systems perspectives, affordance theory [30] and/or the concept of boundary objects [31] could help me understand how teams go about using agile practices and tools differently, depending on their needs and goals, and how this in turn, may reinforce differences through the different action possibilities offered by e.g. technological communication tools, meetings and physical artefacts used in agile activities [32].

[1] A full overview of research results from this line of research is beyond the scope of this paper. See for instance [24], an overview of research and future directions of RCT.

Getting the theory right is a substantial task in a PhD project, and a task I will direct much attention to in the time to come. However, as I will continue to explore how RCT may inform my research project, I will explore recent literature combining RCT with approaches taking into account the role of the technology itself. For instance, Clagett and Karahanna [26] explore the role of relational coordination in digitally mediated work processes and focus in particular on distributed information exchanges for dependency management and the role of boundary spanners in facilitating digitally mediated coordination. Bozan [25] applied RCT in an empirical investigation of collaboration and creative group problem solving in a virtual, distributed team environment and found that RCT's elements of high-quality relationships and high-quality communication did have a positive impact on creative problem-solving in distributed teams.

In the further development of my PhD project, I will look into these and other theoretical approaches to identify the best suited approach to understanding coordination in large-scale settings.

3 Research Objectives and Preliminary Research Questions

The main objective of the project includes identifying success criteria for coordination, such as how to handle interdependencies, enable good communication and better autonomous team-work processes in large-scale agile software development projects. The final output will be a dissertation in the form of an article collection with conference and journal papers.

To gain more understanding about the topics outlined above, I will explore in a field setting research questions such as:

- How are coordination mechanisms used in and across large-scale agile software development projects?
- How do Product Owners coordinate work in large-scale agile software development [15]?
- Which interdependencies operates in and across teams in agile software development projects and what challenges do they pose for team efficiency?
- What is the role of written communication in large-scale agile coordination?

Some of these research questions may be too broad in their current form. Therefore, they will be reworked as the empirical studies are conducted.

4 Research Design

To address the research questions, I primarily plan to use qualitative research methods in a longitudinal case study. The case study approach was chosen because case studies provide depth and detailed knowledge [33] and there is little research-based knowledge about how POs coordinate work in large-scale agile. Data will be collected in the field from several companies associated with the Autonomous teams-project (A-teams) in collaboration with SINTEF.

The data collection methods will include participant observation, individual and potentially group interviews, document analysis [34] and surveys [35]. Collecting a rich data material that can be analyzed in different ways to gain a broad understanding of the research topic and to address the outlined research questions.

4.1 Case Description

I have conducted field work in a large-scale agile development program, referred to as the PubTrans program, since September 2018. The data so far has been collected from a large-scale case in which almost the whole development program is co-located and working with agile development methods. The program started in 2016 and aims to develop a new platform supporting public transportation.

The PubTrans program has thirteen development teams ranging between five and fourteen team members working toward developing the same products. Each team is responsible for their part of the overall product. The PubTrans program can thus be classified as very large-scale agile [12, 36]. In order to coordinate work within and across teams, the program makes use of various electronic tools, such as Slack, Jira, and Confluence; material artefacts such as task boards; and various scheduled and unscheduled meetings. The development teams may choose freely how they solve their tasks and may rely on agile methods of choice. As such, there is no one unified agile approach across the teams.

I spend 1–2 days a week there, observing how they work and attend in particular inter-team meetings. In addition, twelve interviews were conducted in October 2018, with a focus on the Product Owner role, and one interview with a team leader was conducted April 2019. More interviews, with more roles, are planned the coming fall. In addition, I have access to a wide range of written documentation, including Slack logs, Confluence pages and company wiki.

Based on the data collected so far, one conference paper has been presented and published [15]. This paper explores through an RCT lens how Product Owners coordinate within and across agile software development teams in a large-scale public sector program in Norway. Data collection in this program will continue throughout the PhD research project, with supplemental data collection in other companies to follow at a later stage.

In addition to my own presence at the PubTrans program site, one of my supervisors are taking an active part in the fieldwork conducted there. In collaboration, we make sure to provide the program with regular feedback and keep them well informed about the research progress. Whenever a paper is written and sent for review, they are given opportunity to review and approve the data used and results presented, and are offered opportunities to contribute also in terms of co-authoring. Nurturing a good relationship with the case organization is seen as highly valuable for both parties.

All in all, the PubTrans program proves an increasingly valuable case to work with. My access to data is good, and the processes and changes we observe them doing proves interesting and worthwhile of continuous focus. Initially, I planned to include at least three company cases, devoting approximately the same amount of time and efforts to each of them. However, over the past months I have decided that the PubTrans program should serve as the main case for a longitudinal, in-depth case study which can provide rich empirical insights into the research topic [33].

Despite the advantages of longitudinal case studies, there is an inherent trade-off in terms of potential lack of generalizability to other companies and settings [33, 37]. Here, researchers need to weigh the benefits and disadvantages against each other in making a decision. As one means to improve generalizability to other settings, towards the end of the data collection, I will collect supplemental data from other companies I have access to. This data collection will not be as detailed as that of the PubTrans program, however it may serve to cross-check some of the observations and findings into other settings to see if there are great similarities or differences from the PubTrans program to other settings. Of course, no two organizations are alike, so differences are likely to observed. Nevertheless, it is thought worthwhile to do such additional data collection to strengthen my findings.

4.2 Validity Issues and How to Address Them

In terms of validity, which threats and how to control them depends on the research method. For qualitative methods, researcher bias is important to address. I intend to rely on data triangulation, using both interviews, observation and document analysis. Triangulation generate more substantial data, addressing the topics under study from different angels [33]. Further, the analysis of the qualitative data material will involve textual coding. I will use programs such as Nvivo 12 for organizing the codes and conducting the analyses, however, bias and validity threats are prevalent when coding data. Own preconceptions on behalf of the researcher and fatigue are only two of these threats. Further, as much of the analyses will be conducted at least partly in collaboration with others, I will make sure to assess the inter-rater reliability for the analyses to try to assure coding reliability and validity. My supervisors have strong expertise in qualitative research methods, and they will help me ensure validity is sufficiently addressed.

5 Current Research Status and Next Planned Steps

This research project is still in an early phase, and despite the encouraging outset, much remains to be done. In this section, I will describe some of the outstanding issues that should be clarified as I proceed with the research project.

First, the literature on agile software development is already substantial, and the literature on large-scale agile is growing. As I continue to go through these bodies of literatures, I will conduct a literature review to gain a fuller overview of the current state of research on coordination in large-scale agile software development. Here, examining both research papers and the practitioner literature may be a worthwhile endeavor, as there is a substantial practitioner literature within this field.

Second, I will work further on the scope of my research, as it is still somewhat too broad. This includes further delineating the theoretical approaches as well as narrowing down the focus of my research questions. While I will continue exploring the usability of RCT as a theoretical lens for understanding coordination in large-scale agile, I will need to pin down how theory can inform my understanding of how technologies used during development, as well as the product under development, affects coordination.

Finally, I am already planning to collect enough data to allow me to carry on with my research also after the completion of the PhD research project. As described above in Sect. 4, I plan to collect survey data that can be analyzed quantitatively. Much research in the SE field and on large-scale agile is qualitative, which makes it interesting to see whether quantitative research can bring new insights. I have an interest in both types of research. Before starting my PhD, I have also conducted quantitative research based on surveys from another research project. These studies explore distributed, autonomous teams in relation to their coordination under conditions of different levels of initiated and received task interdependence [38] and in relation to how distributed team members perceive certain leadership styles [39]. Continuing such lines of research in a large-scale agile setting could be an interesting future research project.

However, as it can be argued that qualitative studies are more suitable when exploring new grounds [33], I will conduct qualitative research for the PhD project and potentially supplement with quantitative studies at a later stage in my career. In conclusion, doing research on coordination in large-scale agile software development is an exciting endeavor. Many challenges lie ahead as this PhD project continues; however, I remain optimistic about the future and look forward to tackling these challenges as they unfold.

References

1. Dikert, K., Paasivaara, M., Lassenius, C.: Challenges and success factors for large-scale agile transformations: a systematic literature review. J. Syst. Softw. **119**, 87–108 (2016). https://doi.org/10.1016/j.jss.2016.06.013
2. Dingsøyr, T., Moe, N.B.: Research challenges in large-scale agile software development. ACM SIGSOFT Softw. Eng. Not. **38**, 38–39 (2013)
3. Dingsøyr, T., Moe, N.B.: Towards principles of large-scale agile development. In: Dingsøyr, T., Moe, N.B., Tonelli, R., Counsell, S., Gencel, C., Petersen, K. (eds.) XP 2014. LNBIP, vol. 199, pp. 1–8. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-14358-3_1
4. Strode, D.E., Huff, S.L., Hope, B., Link, S.: Coordination in co-located agile software development projects. J. Syst. Softw. **85**, 1222–1238 (2012). https://doi.org/10.1016/j.jss.2012.02.017
5. Dingsøyr, T., Moe, N.B., Seim, E.A.: Coordinating knowledge work in multi-team programs: findings from a large-scale agile development program. Proj. Manag. J. **49**, 64–77 (2018)
6. Lee, G., Xia, W.: Toward agile: an integrated analysis of quantitative and qualitative field data on software development agility. MIS Q. **34**, 87–114 (2010)
7. Moe, N.B., Dingsøyr, T., Dybå, T.: Overcoming barriers to self-management in software teams. IEEE Softw. **26**, 20–26 (2009)
8. Hoda, R., Noble, J., Marshall, S.: Self-organizing roles on agile software development teams. IEEE Trans. Softw. Eng. **39**, 422–444 (2013). https://doi.org/10.1109/TSE.2012.30
9. Hoda, R., Salleh, N., Grundy, J.: The rise and evolution of agile software development. IEEE Softw. **35**, 58–63 (2018). https://doi.org/10.1109/MS.2018.290111318

10. Williams, L., Cockburn, A.: guest editors' introduction: agile software development: it's about feedback and change. Computer **36**, 39–43 (2003)
11. Rigby, D.K., Sutherland, J., Noble, A.: Agile at scale: how to go from a few team to hundreds. Harvard Bus. Rev. **96**, 88–96 (2018)
12. Dingsøyr, T., Nerur, S., Balijepally, V., Moe, N.B.: A decade of agile methodologies: towards explaining agile software development. J. Syst. Softw. **85**, 1213–1221 (2012). https://doi.org/10.1016/j.jss.2012.02.033
13. Jørgensen, M.: Do agile methods work for large software projects? In: Garbajosa, J., Wang, X., Aguiar, A. (eds.) XP 2018. LNBIP, vol. 314, pp. 179–190. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91602-6_12
14. Paasivaara, M., Behm, B., Lassenius, C., Hallikainen, M.: Large-scale agile transformation at Ericsson: a case study. Empirical Softw. Eng. **23**(5), 2250–2596 (2018)
15. Berntzen, M., Moe, N.B., Stray, V.: The product owner in large-scale agile: an empirical study through the lens of relational coordination theory. In: Kruchten, P., Fraser, S., Coallier, F. (eds.) XP 2019. LNBIP, vol. 355, pp. 121–136. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19034-7_8
16. Putta, A., Paasivaara, M., Lassenius, C.: Benefits and challenges of adopting the scaled agile framework (SAFe): preliminary results from a multivocal literature review. In: Kuhrmann, M., et al. (eds.) PROFES 2018. LNCS, vol. 11271, pp. 334–351. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03673-7_24
17. Malone, T.W., Crowston, K.: The interdisciplinary study of coordination. ACM Comput. Surv. (CSUR) **26**, 87–119 (1994)
18. Dingsøyr, T., Bjørnson, F.O., Moe, N.B., Rolland, K., Seim, E.A.: Rethinking coordination in large-scale software development. Presented at the Proceedings of the 11th International Workshop on Cooperative and Human Aspects of Software Engineering (2018)
19. Paasivaara, M., Lassenius, C., Heikkila, V.T.: 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, Lund, Sweden (2012)
20. Nyrud, H., Stray, V.: Inter-team coordination mechanisms in large-scale agile. Presented at the Proceedings of the XP2017 Scientific Workshops (2017)
21. Howison, J., Rubleske, J., Crowston, K.: Coordination theory: a ten-year retrospective. In: Human-Computer Interaction and Management Information Systems: Foundations, pp. 134–152. Routledge (2015)
22. Moe, N.B., Dingsøyr, T., Dybå, T.: A teamwork model for understanding an agile team: a case study of a Scrum project. Inf. Softw. Technol. **52**, 480–491 (2010). https://doi.org/10.1016/j.infsof.2009.11.004
23. Gittell, J.H.: Relational coordination: coordinating work through relationships of shared goals, shared knowledge and mutual respect. In: Relational Perspectives in Organizational Studies: A Research Companion, pp. 74–94 (2006)
24. Gittell, J.H.: New directions for relational coordination theory. In: Spreitzer, G.M., Cameron, K.S. (eds.) The Oxford Handbook of Positive Organizational Scholarship. Oxford University Press (2012)
25. Bozan, K.: The perceived level of collaborative work environment's effect on creative group problem solving in a virtual and distributed team environment. Presented at the Proceedings of the 50th Hawaii International Conference on System Sciences (2017)
26. Claggett, J.L., Karahanna, E.: Unpacking the structure of coordination mechanisms and the role of relational coordination in an era of digitally mediated work processes. Acad. Manag. Rev. **43**, 704–722 (2018)

27. Sebastian, I.M., Bui, T.: The influence of IS affordances on work practices in health care: a relational coordination approach. Presented at the International Conference on Information Systems (2012)
28. Gittell, J.H.: Coordinating mechanisms in care provider groups: relational coordination as a mediator and input uncertainty as a moderator of performance effects. Manag. Sci. **48**, 1408–1426 (2002)
29. Gittell, J.H., Douglass, A.: Relational bureaucracy: structuring reciprocal relationships into roles. Acad. Manag. Rev. **37**, 709–733 (2012)
30. Volkoff, O., Strong, D.M.: Affordance theory and how to use it in IS research. In: The Routledge Companion to Management Information Systems. Routledge, New York (2018)
31. Doolin, B., McLeod, L.: Sociomateriality and boundary objects in information systems development. Eur. J. Inf. Syst. **21**, 570–586 (2012). https://doi.org/10.1057/ejis.2012.20
32. Leonardi, P.M.: When flexible routines meet flexible technologies: affordance, constraint, and the imbrication of human and material agencies. MIS Q. **35**, 147–167 (2011)
33. Yin, R.K.: Case Study Research and Applications: Design and Methods. Sage Publications, Thousand Oaks (2018)
34. Crang, M., Cook, I.: Doing Ethnographies. Sage, Thousand Oaks (2007)
35. Alreck, P.L., Settle, R.B.: The Survey Research Handbook. McGraw-Hill, New York (1994)
36. Dingsøyr, T., Moe, N.B., Fægri, T.E., Seim, E.A.: Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation. Empirical Softw. Eng. 1–31 (2017). https://doi.org/10.1007/s10664-017-9524-2
37. Diefenbach, T.: Are case studies more than sophisticated storytelling?: Methodological problems of qualitative empirical research mainly based on semi-structured interviews. Qual. Quant. **43**, 875 (2009)
38. Berntzen, M., Wong, S.I.: Coordination in distributed, self-managing work teams: the roles of initiated and received task interdependence. In: Proceedings of the 52nd Hawaii International Conference on System Sciences (2019)
39. Wong, S.I., Berntzen, M.: Transformational leadership and leader–member exchange in distributed teams: the roles of electronic dependence and team task interdependence. Comput. Hum. Behav. **92**, 381–392 (2019). https://doi.org/10.1016/j.chb.2018.11.032