

2017

Coordination in Distributed Agile Software Development: A Systematic Review

A B M Nurul Afser Talukder
Auckland University of Technology, a.talukder@aut.ac.nz

Mali Senapathi
Auckland University of Technology, mali.senapathi@aut.ac.nz

Jim Buchan
Auckland University of Technology, jim.buchan@aut.ac.nz

Follow this and additional works at: <https://aisel.aisnet.org/acis2017>

Recommended Citation

Talukder, A B M Nurul Afser; Senapathi, Mali; and Buchan, Jim, "Coordination in Distributed Agile Software Development: A Systematic Review" (2017). *ACIS 2017 Proceedings*. 46.
<https://aisel.aisnet.org/acis2017/46>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2017 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Coordination in Distributed Agile Software Development: A Systematic Review

A.B.M. Nurul Afser Talukder

School of Engineering, Computer and Mathematical Sciences
Department of IT and Software Engineering
Auckland University of Technology
Email: a.talukder@aut.ac.nz

Mali Senapathi

School of Engineering, Computer and Mathematical Sciences
Department of IT and Software Engineering
Auckland University of Technology
Email: mali.senapathi@aut.ac.nz

Jim Buchan

School of Engineering, Computer and Mathematical Sciences
Department of IT and Software Engineering
Auckland University of Technology
Email: jim.buchan@aut.ac.nz

Abstract

Agile methods incorporate many techniques that support coordination in co-located software development teams. However, these benefits do not necessarily transfer to a distributed context. Even though research on coordination in distributed agile software development is growing, there is limited rigorous research on its application in context. Further the extant literature is fragmented, with little cohesive building of cumulative knowledge on coordination in distributed agile software development. This study investigates the scientific evidence between 2006 and 2016 by conducting a systematic review of the literature on coordination in distributed agile software development. The search strategy resulted in 178 studies, of which 50 were identified as primary studies relevant to this research. The studies were classified using three high-level categories: (i) theoretical foundation and application, (ii) tools and techniques, and (iii) challenges. This study provides a structured overview of the current state of knowledge on coordination in distributed agile development, and identifies opportunities for future research.

Keywords coordination, distributed agile software development.

1 Introduction

Coordination in software development involves a collective and effective work approach where two or more people perform interdependent activities towards achieving a common goal (Kraut and Streeter 1995). Coordination has long been recognized as contributing to successful systems development projects both in co-located (Curtis et al. 1988; Kraut and Streeter 1995), and more recently in geographically distributed contexts (Cummings et al. 2009; Strode 2014b). Well-coordinated development offers a number of benefits such as shorter development time frames, well-integrated output that can be produced at lower cost (Andres and Zmud 2002; Espinosa et al. 2007).

Agile methods which are particularly concerned with group endeavour, incorporate useful techniques and mechanisms that support coordination in co-located teams. However, the benefits of coordination mechanisms in co-located agile development do not necessarily transfer to distributed environments (Alzoubi et al. 2016; Hossain et al. 2009a). Over the last decade, triggered by contingencies such as faster time-to-market, diverse competencies, and dynamic business requirements (Saxena et al. 2016), organizations are increasingly using agile methods in distributed environments (Jalali and Wohlin 2012; Strode 2014b). In addition to the challenges associated with spatial, temporal, and configurational differences, projects in distributed settings face a number of specific challenges such as lack of shared understanding, delay in management of project artefacts, conflicts due to misalignment of work, reduced trust, and increased coordination complexity due to miscommunication (Bannerman et al. 2011).

Despite the growing popularity of agile software development in distributed settings, this study identifies two main shortcomings in the literature in this regard. Firstly, although much research has been done on coordination in co-located teams in general (Strode 2014a; Strode and Huff 2015), research focusing specifically on coordination in distributed agile is limited. This is further evidenced in considering the systematic literature reviews (discussed in section 2) published until 2016. These focus either on agile distributed development (coordination is not the focus) or on coordination in distributed development (agile is not the focus). Secondly, there is no comprehensive view of the current state of knowledge on coordination in distributed agile software development (DASD). The current knowledge is fragmented among research on coordination theories, distributed software development and agile software development.

To address this gap in knowledge, the goal of this study is to conduct a systematic review on coordination in distributed agile software development. Conducting a systematic review is important as it provides a useful classification and structured overview of the current research on coordination in DASD. It can be used to provide a valuable baseline to assist new research efforts (Kitchenham et al. 2010). The aim of this study is to answer the following questions:

- 1) what knowledge areas in coordination in distributed agile software development are addressed?
- 2) what types of research are published in the area of coordination?

The remainder of the paper is organized as follows: section 2 provides the background and related work, and section 3 presents the research methodology. The results and findings are presented in section 4. In section 5, conclusions and directions for future research are presented.

2 Background and related work

This section commences with coordination in different agile development contexts. A summary of related reviews conducted in distributed development contexts is then discussed. Related work on coordination in DASD is also discussed.

2.1 Coordination in Agile Software Development

As agile methods continue to grow in popularity, research on coordination in agile software development also emerged. Cao and Ramesh (2007) identified specific consistencies between the coordination modes of organisation theory (e.g. personal, group) (Van De Ven et al. 1976) and coordination mechanisms of agile practices (e.g. co-located customers, short iterations). By applying Malone and Crowston (1994)'s coordination theory, Pikkarainen et al. (2008) found that practices such as sprint planning meetings, open office space, and daily meetings promote communication in small co-located projects. These findings are also supported by more recent research, where artefacts such as product backlog, sprint backlog, and burn-down charts have been identified as effective coordination mechanisms (Wagenaar et al. 2015). Strode et al. (2012) developed a theory of coordination which proposed that the

coordination effectiveness of an agile software development project is affected by the coordination strategy of the project.

2.2 Coordination in Distributed Software Development

Organisations are increasingly using distributed development to deliver high-quality software to distributed users and customers at lower development costs (Powell et al. 2004). However, factors such as geographic distance, time, organisational boundaries, and functional boundaries inherent in distributed contexts represent significant barriers for distributed software development (Lee et al. 2006). With this, a number of coordination problems have become known. The exacerbation of coordination problems due to the significant differences between co-located and distributed development has been well recognised in the literature (Ågerfalk et al. 2006; Espinosa et al. 2007; Herbsleb 2007; Herbsleb and Grinter 1999). In a recent systematic literature review of 101 selected studies, 35% of studies identified lack of coordination as one of the major challenges (Niazi et al. 2016)..

2.3 Coordination in Agile Distributed Software Development

Though originally intended for small co-located projects, agile methods are now being increasingly used in distributed environments (Jalali and Wohlin 2012; Strode 2014b). In order to cope with the current dynamic business requirements, software development organisations are striving to blend agile software development methods such as Scrum and distributed development to reap the benefits of both (Ramesh et al. 2006). However, agile and distributed development approaches are based significantly different tenets (Ramesh et al. 2006). For example, while agile methods mainly rely on informal processes to facilitate coordination, distributed development relies on formal mechanisms to coordinate their tasks. These differences can contribute to specific coordination issues and challenges in addition to those associated with the inherent characteristics of distributed development such as spatial, temporal and functional differences. Nevertheless, various coping strategies and solutions have been proposed to address these differences, and several software development organisations have reported successful implementations of agile in distributed contexts (Šmite et al. 2010).

By applying Mintzberg's theory of coordination, (Hossain et al. 2009c) identified that specific agile practices such as Sprint planning, Sprint Review and Retrospective meetings provided significant support to coordination and building positive relationships between the distributed team members. Using a multi-team systems concept and drawing upon existing coordination literature, Scheerer et al. (2014) investigated the time dependent interplay between coordination type, locus and direction and its key contingencies in a process theoretic approach. By drawing upon Organisational Discontinuity Theory, Crowston et al. (2016) developed a model to examine the effects of discontinuities on inter-team coordination in large-scale agile software development. Other studies (Alyahya et al. 2011; Feiner 2016) have investigated the role of tools and technologies including more recent advancements such as augmented reality (Lukosch et al. 2015) and Virtual Scrum (Rodriguez et al. 2015) in facilitating coordination in DASD.

2.4 Related Systematic Reviews

Prior to conducting this study, previous systematic literature reviews published until 2016 were reviewed to ensure that the research questions of this study were not been answered before. Table 1 shows the results of the previous reviews. Out of the 9 reviews, 5 focused on the agile in distributed development, and 4 focused on coordination related aspects in distributed development.

	Title	Reference
1	Using Scrum in Global Software Development: A Systematic Literature Review	(Hossain et al. 2009a)
2	The Lean Gap: A Review of Lean Approaches to Large-Scale Software Systems Development	(Pernstål et al. 2013)
3	Agile Development in Large and Distributed Environments	(Razavi and Ahmad 2014)
4	Agile Global Software Development Communication Challenges: A Systematic Review	(Alzoubi and Gill 2014)
5	Distributed Pair Programming: A Systematic Literature Review	(Estacio & Prikladnicki 2015)

Focus on agile methods and practices in distributed development, but coordination aspects not specifically addressed.

1	Software Configuration Management in Global Software Development: A Systematic Map	(Fauzi et al. 2010)	Some coordination aspects are addressed but agile development is not the focus
2	Tools used in Global Software Engineering: A Systematic Review	(Portillo-Rodriguez et al. 2012)	
3	Empirical Studies on the use of Social Software in Global Software Development: A Systematic Mapping Study	(Giuffrida and Dittrich 2013)	
4	Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature.	(Steinmacher et al. 2013)	

Table 1. Related Systematic Reviews in Distributed Software Development

The above review demonstrate that coordination in distributed software development is a growing area of interest to practitioners and researchers. There is, however, no systematic overview of research on coordination in DASD which can provide a structure to existing knowledge as well as identify gaps in research which need addressing. This motivates the research undertaken in this paper. The next section describes the process to undertake this systematic study.

3 Research Methodology

This section outlines the systematic review process adopted in this study, which follows the established guidelines and procedures proposed in the literature (Dybå and Dingsøy 2008; Kitchenham and Charters 2007; Okoli and Schabram 2010). The review process consists of the following steps: establish research aim and define research questions, conduct search, screening and selection of papers, and data extraction and classification.

3.1 Establish research aim and define research questions

The motivation to conduct a systematic review study is to “identify, analyse, and interpret all available evidence related to a specific question in a way that is unbiased and repeatable” (Kitchenham and Charters 2007). In this research, the motivation is to provide a state-of-the-art of coordination research in distributed agile software between 2006 and 2016. The main objectives of this study are to (i) establish the body of knowledge by identifying and classifying the available research on coordination in DASD, and (ii) identify the main types of research published. To achieve these objectives, the following research questions will be answered:

RQ1: What are the main knowledge areas in coordination in distributed agile software development?

RQ2: What types of research are published in the area of coordination?

3.2 Conduct search

The search strategy included electronic databases and manual searches of conference proceedings. The databases that were searched are: Scopus, ACM Digital Library, IEEE Xplore, and AIS Electronic Library (AISeL). In addition, all volumes of the XP and Agile Development Conference proceedings were searched manually. Search strings were formulated by combining three key concepts using ‘AND’ operator. Keywords in each key concept were combined using ‘OR’ operator to ensure good coverage of papers related to that particular context area (see Table 2).

‘A’ AND ‘B’ AND ‘C’	
A	Agile OR scrum OR XP OR extreme programming OR lean OR Kanban
B	Global* OR distributed OR disperse* OR “large-scale” OR “large scale”
C	coordination OR “co-ordination”

Table 2. Search terms

Agile methods were limited to the most commonly used methods in practice, i.e. agile, scrum, extreme programming, lean and Kanban, and all types of distributed contexts were included (e.g. large-scale,

global, and dispersed development). The records were imported into Microsoft Excel using the following format (i) title, (ii) author/s, (iii) year, (iv) publication type, and (v) abstract.

3.3 Screening and selection procedure

The search string applied on the four databases (i.e. ACM Digital Library, Scopus, IEEE Xplore, and AISEL) retrieved 178 papers. Two authors independently analysed the 178 publications and applied the inclusion and exclusion criteria. This included excluding studies based on titles and abstracts, removing duplicates, non-English publications, and non-peer reviewed scientific papers. This process resulted in 53 papers being excluded and 125 studies included. Next, all three authors analysed the 125 papers over a two-week period. Both during and at the end of this period, meetings were held to conduct in-depth reviews of each paper. The outcome of this produced 50 papers which were assessed for quality using the criteria proposed by Dybå and Dingsøyr (2008) (Figure 1).

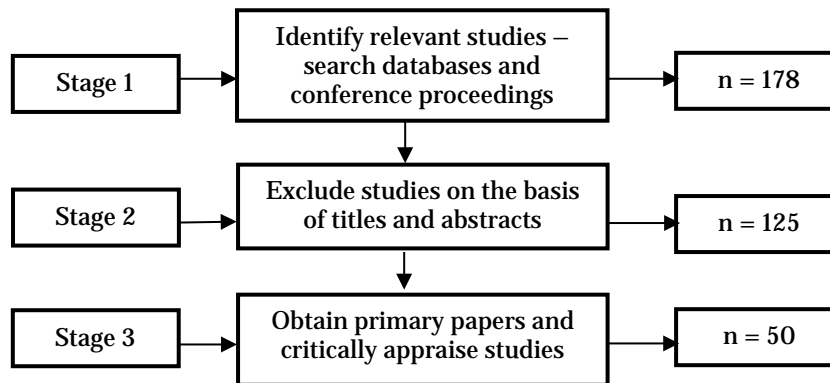


Figure 1: Stages of the study selection process

3.3.1 Inclusion and Exclusion Criteria

Studies were eligible for inclusion in the review if they clearly address specific aspects of coordination in DASD and presented empirical data. Studies published between 2006 and 2016 were included. Only studies written in English were included. Studies were excluded if their main focus was not coordination in DASD. For example, studies that focused on large-scale agile development, but not distributed were excluded. In cases where studies were published in more than one outlet, then the most recent version was included. For example, if there were two studies, one conference and a later journal publication, then the conference publication was excluded. Non peer-reviewed scientific papers (e.g. books, book chapters, experience reports) were excluded.

3.4 Data Extraction and Classification

The derivation of the classification scheme was based on Petersen's classification guidelines (Petersen et al. 2015). First, the abstracts were reviewed to identify the main concepts and themes related to coordination in DASD using the 'keywording' process described in the guideline. Secondly, the set of keywords collected from different papers were combined together to develop a high-level understanding of the theme, context, and contribution of the research in order to form three classification categories: Theoretical Foundation and Application, Techniques and Tools and Challenges (Table 3).

Classification Scheme	Key Reported Aspects
Theoretical Foundation and Application	Theories related to coordination in DASD: application of well-established theories to conceptualize activities, dependencies, dimensions, and effectiveness and coordination roles.
Techniques And Tools	Coordination practices and mechanisms, strategies, artefacts, conceptual frameworks and tools for coordination support
Challenges	Coordination challenges related to DASD

Table 3. Classification Scheme

3.5 Validity and Reliability

Construct validity relates to the alignment of what is investigated to what the researchers had in mind, as well as the completeness of the results. To reduce this threat, a data collection process was designed in advance that defined the research questions and the search strategy. Search terms were clearly defined that would enable us to identify the relevant literature. In addition, we used synonyms of the main terms to capture variations that may appear in the literature. To address the second aspect of construct validity (i.e. completeness of results), ensuring we found all the papers on the topic area of interest, we undertook the search in publication databases that are reputed to be well indexed. *Reliability* (repeatable with the same results) of the data collection was ensured by defining our search terms and procedures in sufficient detail so they can be replicated by other researchers. The inclusion/exclusion criteria are also described to a level of detail that can be replicated by others and have little room for misinterpretation. The categorization is a possible threat to validity and although the procedure is described, it is not certain that others would come up with the same classification schemes. To address this, the classifications were conducted by the first author and validated by the second and third authors. *Internal reliability* is a low threat in this study since only descriptive statistics were used in the analysis of the data. *External validity* is not in question for this study since we do not try to generalise our results to other review studies.

4 Results

This section presents the results of the study using the classification scheme depicted in Table 3 to answer the research questions. Each category is further classified into sub-categories. The aim of the first research question is to review and classify the main knowledge areas on coordination in DASD. The three main areas (Theoretical Foundation and Application, Techniques and Tools, and Challenges) are discussed in the following sub sections.

4.1 Theoretical Foundation and Application:

This category is classified into four sub-categories: Theories of coordination, Coordination types, Coordination effectiveness, and Roles.

4.1.1 Theories of Coordination:

20 out of 50 studies were classified under this sub-category. 9 of these 20 studies used coordination theory (CT) (Malone and Crowston 1994) to understand the main coordination tasks, their dependencies, mechanisms to support dependencies and various actors involved in the coordination process. For example, CT has been used as a theoretical lens to gain a better understanding of the process of formulating coordination strategy (Bick et al. 2014; Xu 2009), and also to understand the role of communication in supporting coordination (Li and Maedche 2012; Modi et al. 2013). 8 out the 20 studies have applied Mintzberg's coordination mechanisms (i.e. direct supervision, standardization & mutual adjustment) to understand their relevance in agile organizational contexts (Hossain 2008; Hossain et al. 2009b; Morken 2014). Other studies have used the Articulation of work theory (Pries-Heje and Pries-Heje 2011a; Pries-Heje and Pries-Heje 2011b) and David Parna's Division of Labor theory (Díaz et al. 2011) to investigate how coordination mechanisms can be used to manage complex distributed projects.

4.1.2 Coordination Types:

6 out of 50 studies focused on different types of coordination types associated with managing coordination dependencies. For example, Bick et al. (2014) identify three types of coordination based on dependency: Task coordination (management of task dependencies), Knowledge coordination (management of dependencies based on allocation and sharing of information) and Technical coordination (management of dependencies in technical architecture and software components). Other studies (Li and Maedche 2012; Prikładnicki and Carmel 2013) have applied the three main types of coordination to understand management of dependencies, i.e., Mechanistic coordination to manage dependencies with plans, routines (e.g., schedules, procedure manuals), Organic coordination to manage dependencies through communication (e.g., providing feedback, mutual adjustment), and Cognitive coordination to coordinate through shared cognition. Coordination is conceptualised based on mode of communication: while Morken (2014) conceptualise coordination as either vertical (via supervisor) or horizontal (one-to-one), Bick et al. (2016) conceptualise using top-down planning (i.e. via mechanistic, centralized and vertical coordination) and bottom-up adjustment (i.e. organic, decentralized and horizontal coordination).

4.1.3 Coordination Effectiveness:

We found only 3 studies that focused on coordination effectiveness in distributed agile software development. Li and Maedche (2012) conceptualise coordination effectiveness as the extent to which the dependencies are managed using three aspects: technical, temporal and process. While Lagerberg et al. (2013) study the impact of agile practices on coordination effectiveness in DASD, Bick et al. (2014) highlight the influence of structural incongruence on coordination effectiveness from a multi-team systems perspective.

4.1.4 Coordination Roles:

We found 3 studies that highlight the need for a specific representative to coordinate the dispersed teams and customer representatives by identifying three specific roles: Coordinator (process facilitator) in agile-rigid environment (Yadav et al. 2007), Technical Area Responsible (TAR) in Functional Feature Teams (Moe et al. 2014), and Scrum Master as coordinator (Bass 2016).

In summary, a number of studies have used the main coordination theories (Malone and Crowston 1994) to investigate the main coordination types, tasks, dependencies, and mechanisms associated with the coordination process in DASD. There is limited research on coordination effectiveness, and in particular, there is apparent lack of specific guidelines for measuring and monitoring coordination effectiveness in DASD. Though some specific roles have been identified, there is no specific research on understanding the functions and obligations associated with these roles.

4.2 Techniques and Tools

This category was further classified into three main perspectives: Mechanisms, Models and Frameworks, and Tools.

4.2.1 Mechanisms:

17 out of 50 studies focused on specific co-ordination mechanisms (i.e. 32% of the results) in DASD. By drawing on Mintzberg's coordination framework 5 of the 17 studies (Hole and Moe 2008; Hossain 2008; Hossain et al. 2009a; Moe et al. 2015; Morken 2014) highlight that while coordination in distributed software development can be supported by formal mechanisms such as by standardization and direct supervision, coordination in DASD is better supported by informal communication and mutual adjustment mechanisms. Specific agile practices that support coordination in DASD such as daily stand-up meetings, collective code ownership, and the use of daily forum of forums meetings (e.g. scrum-of-scrum) have been identified (Hole and Moe 2008; Šāblis and Šmite 2016; Xu 2009). The use of informal communication strategies such as Skype chat and Concurrent Versioning System (CVS) used to develop shared understanding have been recognised (Procter et al. 2011). The use of Communities of Practices and feature coordination (common practices for coordinating teams working on specific features) have also been studied (Bjørnson and Vestues 2016; Paasivaara and Lassenius 2014).

4.2.2 Models & Frameworks:

4 out of 50 studies propose specific frameworks that can be used to improve coordination support in in DASD. While Bergadano et al. (2014) propose the SCoAP (Support for Communication and Agile Project Management) framework, Díaz et al. (2011) demonstrate how a Feature Partitioning Method (FPM) can be used to 1) reduce intensive communication and coordination, 2) provide better utilization of resources, and 3) produce higher quality features. Morken (2014) examine how critical path diagram can be used to improve team coordination by mapping the known coordination dependencies. And, Bass (2016) investigate how the use of specific artefacts such as Release code binaries (what), release plan (when), integration test (how), and product backlog (what) can be used to facilitate team coordination in agile contexts.

4.2.3 Tools:

14 out of 50 studies investigated how tools can be used to support various aspects of the coordination process in distributed agile development. Specific tools to address the limitations of current progress tracking systems (e.g. Rally, VersionOne) in coordinating the impact of the technical factors (e.g. Unit and Acceptance Testing, Source code versioning) (Alyahya et al. 2011; Feiner 2016) have been proposed: while Mak and Kruchten (2006) propose the NextMove tool to support asynchronous task prioritization and allocation, Alyahya et al. (2011) present a progress management tool to support the coordination of technical activities in progress, and Feiner (2016) propose ScrumPy, a Scrum based tool to manage and coordinate requirements.

In summary, a number of studies have contributed to understanding the significance of various coordination mechanisms in DASD. Few studies have proposed specific frameworks that can provide improved coordination support. While many studies have investigated the role of tools in supporting various aspects of coordination, most of these are in their initial proposal and are yet to be evaluated in real-life scenarios.

4.3 Challenges

Almost half of the selected studies (24 out of 50 papers) focused on identifying and addressing the main coordination challenges in distributed software development. Most of these studies (20 out of 24 studies) highlight challenges related to distance (i.e. geographical, temporal & socio-cultural) (Bannerman et al. 2011; Hossain 2008; Hossain et al. 2009a). Some studies have identified the interrelation and overlap between some of these challenges. For example, Hossain et al. (2009c) highlight how communication challenges can lead to additional challenges in project management, and team cohesion and bonding. Other challenges associated with how coordination mechanisms (i.e. standardization, direct supervision, and mutual adjustment) can impact challenges related to geographical, temporal and socio-cultural distance factors have also been investigated (Niazi et al. 2016).

4.4 Types of research

To answer the second research, the classification suggested by Wieringa et al. (2006) was used to classify the studies based on research type: Evaluation research, Validation research, Solution Proposal, Philosophical paper, Opinion paper and Experience paper (see Figure 2). The results show that 60% (31 out of 50 studies) of them used evaluation type research (e.g. application and evaluation of agile practices, coordination techniques). 17% of the studies proposed specific solutions in terms of either tools or frameworks, i.e. Solution Proposal. Other types of research were: Philosophical papers (11%), Opinion papers (6%), Experience reports (6%) and Validation research (0%).

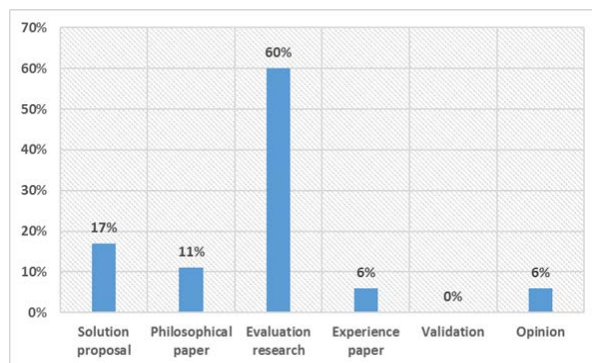


Figure 2: Classification of papers based on research type

In terms of the research methodology, more than half of the selected studies used Case Study (55.8%) method to either define a new theory, explore or test an existing theory, and to explain events and activities in real-life context (e.g. a particular organization) (Myers 2013). Other reported methods were: Field study (6%), Mixed Method and Grounded Theory (4% each) and Design science, Quasi-experiment, and Action research (1% each). 21% of the studies did not specify a research methodology as these papers focused on either developing or proposing specific solution proposals (Table 4).

Research Method	Frequency	Percentage
Case Study	29	55.8%
Field Study	3	5.8%
Mixed Method	2	3.8%
Grounded Theory	2	3.8%
Design science	1	1.9%
Quasi-experiment	1	1.9%
Action research	1	1.9%

Table 4. Research Methods Classification

5 Conclusion and directions for future research

Our review shows that the majority of research has been on the application of general coordination theories to understanding various aspects of coordination in DASD, e.g. activities, roles, mechanisms and strategies. However, there is relatively limited research on the effective implementation and management of these aspects in DASD. For example, while few papers highlight the significance of specific roles on effective coordination in DASD, there is no specification of the functions and obligations associated with these roles. Some studies have contributed to gaining a better understanding of coordination effectiveness and strategy formulation in general; however, there is an apparent lack of specific guidelines on strategy formulation process, and measuring and monitoring coordination effectiveness in DASD. This indicates a need to investigate not only the impact of specific practices, activities, dependencies and strategies, but also to explore the relationship between the various elements and their impact on coordination effectiveness in DASD. It is also important to note future research on DASD must adopt a tradition of cumulative building of knowledge in order to contribute to theoretical development. We hope our review and classification scheme will help researchers to position and plan their future research.

6 References

- Ågerfalk, J., Fitzgerald, B., and In, O. P. 2006. "Flexible and Distributed Software Processes: Old Petunias in New Bowls," *Communications of the ACM*: Citeseer.
- Alyahya, S., Ivins, W. K., and Gray, W. A. 2011. "Co-Ordination Support for Managing Progress of Distributed Agile Projects," *Proceedings - 2011 6th IEEE International Conference on Global Software Engineering Workshops, ICGSE Workshops 2011*), pp. 31-34.
- Alzoubi, Y. I., and Gill, A. Q. 2014. "Agile Global Software Development Communication Challenges: A Systematic Review," *Proceedings of Pacific Asia Conference on Information Systems*), pp. 1-13.
- Alzoubi, Y. I., Gill, A. Q., and Al-Ani, A. 2016. "Empirical Studies of Geographically Distributed Agile Development Communication Challenges: A Systematic Review," *Information & Management* (53:1), pp. 22-37.
- Andres, H. P., and Zmud, R. W. 2002. "A Contingency Approach to Software Project Coordination," *Journal of Management Information Systems* (18:3), pp. 41-70.
- Bannerman, P. L., Hossain, E., and Jeffery, R. 2011. "Scrum Practice Mitigation of Global Software Development Coordination Challenges: A Distinctive Advantage?," *Proceedings of the Annual Hawaii International Conference on System Sciences*), pp. 5309-5318.
- Bass, J. M. 2016. "Artefacts and Agile Method Tailoring in Large-Scale Offshore Software Development Programmes," *Information and Software Technology* (75), pp. 1-16.
- Bergadano, F., Bosio, G., and Spagnolo, S. 2014. "Supporting Collaboration between Customers and Developers: A Framework for Distributed, Agile Software Development," *International Journal of Distributed Systems and Technologies* (5:2), pp. 1-16.
- Bick, S., Scheerer, A., and Spohrer, K. 2016. "Inter-Team Coordination in Large Agile Software Development Settings: Five Ways of Practicing Agile at Scale," *Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops*), pp. 1-5.
- Bick, S., Scheerer, A., Spohrer, K., and Kude, T. 2014. "Software Development in Multiteam Systems: A Longitudinal Study on the Effects of Structural Incongruences on Coordination Effectiveness," ... *Pre-ICIS International Research Workshop on ...:Irwitpm*), pp. 1-56.
- Bjørnson, F. O., and Vestues, K. 2016. "Knowledge Sharing and Process Improvement in Large-Scale Agile Development," New York, NY, USA: ACM, pp. 7:1--7:5.
- Cao, L., and Ramesh, B. 2007. "Agile Software Development: Ad Hoc Practices or Sound Principles?," *IT Professional* (9:2), pp. 41-47.
- Crowston, K., Chudoba, K., Watson-Manheim, M. B., and Rahmati, P. 2016. "Inter-Team Coordination in Large-Scale Agile Development: A Test of Organizational Discontinuity Theory," *Proceedings of the Scientific Workshop Proceedings of XP2016*: ACM, p. 2.
- Cummings, J. N., Espinosa, J. A., and Pickering, C. K. 2009. "Crossing Spatial and Temporal Boundaries in Globally Distributed Projects: A Relational Model of Coordination Delay," *Information Systems Research* (20:3), pp. 420-439.
- Curtis, B., Krasner, H., and Iscoe, N. 1988. "A Field Study of the Software Design Process for Large Systems," *Communications of the ACM* (31:11), pp. 1268-1287.
- Díaz, J., Garbajosa, J., and Pérez, J. 2011. "A Feature Partitioning Method for Distributed Agile Release Planning," *Lecture Notes in Business Information Processing* (77:May 2011), pp. 318-319.
- Dybå, T., and Dingsøyr, T. 2008. "Empirical Studies of Agile Software Development: A Systematic Review," *Information and software technology* (50:9), pp. 833-859.

- Espinosa, J. A., Slaughter, S. A., Kraut, R. E., and Herbsleb, J. D. 2007. "Team Knowledge and Coordination in Geographically Distributed Software Development," *Journal of Management Information Systems* (24:1), pp. 135-169.
- Fauzi, S. S. M., Bannerman, P. L., and Staples, M. 2010. "Software Configuration Management in Global Software Development: A Systematic Map," *2010 Asia Pacific Software Engineering Conference* (3:1), pp. 404-413.
- Feiner, M. E. 2016. "How Scrum Tools May Change Your Agile Software Development Approach."
- Giuffrida, R., and Dittrich, Y. 2013. "Empirical Studies on the Use of Social Software in Global Software Development - a Systematic Mapping Study," *Information and Software Technology* (55:7), pp. 1143-1164.
- Herbsleb, J. D. 2007. "Global Software Engineering: The Future of Socio-Technical Coordination," *2007 Future of Software Engineering*; IEEE Computer Society, pp. 188-198.
- Herbsleb, J. D., and Grinter, R. E. 1999. "Architectures, Coordination, and Distance: Conway's Law and Beyond," *IEEE software* (16:5), pp. 63-70.
- Hole, S., and Moe, N. B. 2008. "A Case Study of Coordination in Distributed Agile Software Development," *Software Process Improvement - Proceedings of the 15th European Conference European System & Software Process Improvement and Innovation, EuroSPI 2008, Dublin, Ireland, September 3-5, 2008*:June), pp. 189-200.
- Hossain, E. 2008. "Coordinating Mechanisms for Agile Global Software Development," *Proceedings - 2008 3rd IEEE International Conference Global Software Engineering, ICGSE 2008*, pp. 257-263.
- Hossain, E., Babar, M. A., and Paik, H.-y. 2009a. "Using Scrum in Global Software Development: A Systematic Literature Review," *SBIE - Simpósio Brasileiro de Informática na Educação*, pp. 175-184.
- Hossain, E., Babar, M. A., and Verner, J. 2009b. "How Can Agile Practices Minimize Global Software Development Co-Ordination Risks?."
- Hossain, E., Babar, M. A., and Verner, J. 2009c. "How Can Agile Practices Minimize Global Software Development Co-Ordination Risks?," *Communications in Computer and Information Science* (42), pp. 81-92.
- Jalali, S., and Wohlin, C. 2012. "Global Software Engineering and Agile Practices: A Systematic Review," *Journal of software: Evolution and Process* (24:6), pp. 643-659.
- Kitchenham, B., and Charters, S. 2007. "Guidelines for Performing Systematic Literature Reviews in Software Engineering," *Engineering EBSE Technical Report EBSE-2007-01*.
- Kitchenham, B., Pretorius, R., Budgen, D., Brereton, O. P., Turner, M., Niazi, M., and Linkman, S. 2010. "Systematic Literature Reviews in Software Engineering—a Tertiary Study," *Information and Software Technology* (52:8), pp. 792-805.
- Kraut, R. E., and Streeter, L. A. 1995. "Coordination in Software Development," *Communications of the ACM* (38:3), pp. 69-82.
- Lagerberg, L., Skude, T., Emanuelsson, P., Sandahl, K., and Ståhl, D. 2013. "The Impact of Agile Principles and Practices on Large-Scale Software Development Projects: A Multiple-Case Study of Two Projects at Ericsson," pp. 348-356.
- Lee, G., DeLone, W., and Espinosa, J. A. 2006. "Ambidextrous Coping Strategies in Globally Distributed Software Development Projects," *Communications of the ACM* (49:10), pp. 35-40.
- Li, Y., and Maedche, A. 2012. "Formulating Effective Coordination Strategies in Agile Global Software Development Teams," *33rd International Conference on Information Systems*, pp. 1-12.
- Lukosch, S., Billinghamurst, M., Alem, L., and Kiyokawa, K. 2015. "Collaboration in Augmented Reality," *Computer Supported Cooperative Work (CSCW)* (24:6), pp. 515-525.
- Mak, D. K. M., and Kruchten, P. B. 2006. "Task Coordination in an Agile Distributed Software Development Environment," *European Conference on Software Process Improvement*:May), pp. 606-611.
- Malone, T. W., and Crowston, K. 1994. "The Interdisciplinary Study of Coordination," *ACM Computing Surveys* (26:1), pp. 87-119.
- Modi, S., Abbott, P., and Counsell, S. 2013. "Raising Awareness in Distributed Agile Development - a Case Study Perspective," *UK Academy for Information Systems Conference Proceedings 2013*, pp. 1-26.
- Moe, N. B., Cruzes, D. S., Dybå, T., and Engebretsen, E. 2015. "Coaching a Global Agile Virtual Team," pp. 33-37.
- Moe, N. B., Šmite, D., Šablīs, A., Börjesson, A.-L., and Andréasson, P. 2014. "Networking in a Large-Scale Distributed Agile Project," New York, NY, USA: ACM, pp. 12:11--12:18.
- Morken, R. A. T. 2014. "Coordination in Large-Scale Agile Development," (May).
- Myers, M. D. 2013. *Qualitative Research in Business and Management*. Sage.
- Niazi, M., Mahmood, S., Alshayeb, M., Riaz, M. R., Faisal, K., Cerpa, N., Khan, S. U., and Richardson, I. 2016. "Challenges of Project Management in Global Software Development: A Client-Vendor Analysis," *Information and Software Technology* (80), pp. 1-19.
- Okoli, C., and Schabram, K. 2010. "A Guide to Conducting a Systematic Literature Review of Information Systems Research,").

- Paasivaara, M., and Lassenius, C. 2014. "Communities of Practice in a Large Distributed Agile Software Development Organization - Case Ericsson," *Information and Software Technology* (56:12), pp. 1556-1577.
- Pernstål, J., Feldt, R., and Gorschek, T. 2013. "The Lean Gap: A Review of Lean Approaches to Large-Scale Software Systems Development," *Journal of Systems and Software* (86:11), pp. 2797-2821.
- Petersen, K., Vakkalanka, S., and Kuzniarz, L. 2015. "Guidelines for Conducting Systematic Mapping Studies in Software Engineering: An Update," *Information and Software Technology* (64), pp. 1-18.
- Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., and Still, J. 2008. "The Impact of Agile Practices on Communication in Software Development," *Empirical Software Engineering* (13:3), pp. 303-337.
- Portillo-Rodriguez, J., Vizcaino, A., Piattini, M., and Beecham, S. 2012. "Tools Used in Global Software Engineering: A Systematic Mapping Review," *Information and Software Technology* (54:7), pp. 663-685.
- Powell, A., Piccoli, G., and Ives, B. 2004. "Virtual Teams: A Review of Current Literature and Directions for Future Research," *ACM Sigmis Database* (35:1), pp. 6-36.
- Pries-Heje, L., and Pries-Heje, J. 2011a. "Agile & Distributed Project Management : A Case Study Revealing Why Scrum Is Useful," *Ecis:2011*, pp. Paper 217-Paper 217.
- Pries-Heje, L., and Pries-Heje, J. 2011b. "Why Scrum Works: A Case Study from an Agile Distributed Project in Denmark and India," pp. 20-28.
- Prikladnicki, R., and Carmel, E. 2013. "Is Time-Zone Proximity an Advantage for Software Development? The Case of the Brazilian It Industry," pp. 973-981.
- Procter, R., Rouncefield, M., Poschen, M., Lin, Y., and Voss, A. 2011. "Agile Project Management: A Case Study of a Virtual Research Environment Development Project," *Computer Supported Cooperative Work* (20:3), pp. 197-225.
- Ramesh, B., Cao, L., Mohan, K., and Xu, P. 2006. "Can Distributed Software Development Be Agile?," *Communications of the ACM* (49:10), pp. 41-46.
- Razavi, A. M., and Ahmad, R. 2014. "Agile Development in Large and Distributed Environments: A Systematic Literature Review on Organizational, Managerial and Cultural Aspects," *MySEC, Malaysian Software Engineering Conference* pp. 216-221.
- Rodriguez, G., Soria, A., and Campo, M. 2015. "Virtual Scrum: A Teaching Aid to Introduce Undergraduate Software Engineering Students to Scrum," *Computer Applications in Engineering Education* (23:1), pp. 147-156.
- Šablīs, A., and Šmite, D. 2016. "Agile Teams in Large-Scale Distributed Context," *Proceedings of the Scientific Workshop Proceedings of XP2016 on - XP '16 Workshops*, pp. 1-5.
- Saxena, A., Venkatagiri, S., and Bandi, R. K. 2016. "Managing Inherent Conflicts in Agile Distributed Development: Evidence from Product Development,").
- Scheerer, A., Hildenbrand, T., and Kude, T. 2014. "Coordination in Large-Scale Agile Software Development: A Multiteam Systems Perspective," pp. 4780-4788.
- Šmite, D., Wohlin, C., Gorschek, T., and Feldt, R. 2010. "Empirical Evidence in Global Software Engineering: A Systematic Review," *Empirical software engineering* (15:1), pp. 91-118.
- Steinmacher, I., Chaves, A. P., and Gerosa, M. A. 2013. "Awareness Support in Distributed Software Development: A Systematic Review and Mapping of the Literature," *Computer Supported Cooperative Work* (22:2-3), pp. 113-158.
- Strode, D. 2014a. "Measuring Coordination in Agile Software Development," *Management*, p. 2.
- Strode, D. 2014b. "Measuring Coordination in Agile Software Development," *International Research Workshop on Information Project Management (IRWITPM)*, Auckland, New Zealand, pp. 37-48.
- Strode, D. E., Huff, S. L., Hope, B., and Link, S. 2012. "Coordination in Co-Located Agile Software Development Projects," *Journal of Systems and Software* (85:6), pp. 1222-1238.
- Strode, D. E. a., and Huff, S. L. b. c. 2015. "A Coordination Perspective on Agile Software Development," *Modern Techniques for Successful IT Project Management*, pp. 64-96.
- Van De Ven, A. H., Delbecq, A. L., and Koenig Jr, R. 1976. "Determinants of Coordination Modes within Organizations," *American Sociological Review* (41:2), pp. 322-338.
- Wagenaar, G., Helms, R., Damian, D., and Brinkkemper, S. 2015. "Artefacts in Agile Software Development," *International Conference on Product-Focused Software Process Improvement*: Springer, pp. 133-148.
- Wieringa, R., Maiden, N., Mead, N., and Rolland, C. 2006. "Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and a Discussion,").
- Xu, P. 2009. "Coordination in Large Agile Projects," *The Review of Business Information Systems* (13:4), pp. 29-43.
- Yadav, V. a., Adya, M. b., Nath, D. a., and Sridhar, V. a. 2007. "Investigating an 'Agile-Rigid' Approach in Globally Distributed Requirements Analysis."

Copyright

Copyright: © 2017 Talukder, Senapathi & Buchan. This is an open-access article distributed under the terms of the [Creative Commons Attribution-NonCommercial 3.0 Australia License](https://creativecommons.org/licenses/by-nc/3.0/au/), which permits non-

commercial use, distribution, and reproduction in any medium, provided the original author and ACIS are credited.