

Challenges in Distributed Agile Software Development Environment: A Systematic Literature Review

**Imran Ghani^{1*}, Angelica Lim², Muhammad Hasnain², Israr Ghani³
and Muhammad Imran Babar⁴**

¹Indiana University of Pennsylvania, USA
[e-mail: ighani@iup.edu]

²Monash University Malaysia
[email: {alim0004, muhammad.malik1} @student.monash.edu]

³Faculty of Computing, Universiti Teknologi Malaysia
[e-mail: israrghani@outlook.com]

⁴Army Public College of Management & Sciences, Pakistan
[e-mail: imran.babar77@gmail.com]

*Corresponding author: Imran Ghani

Received January 17, 2019; accepted March 4, 2019; published September 30, 2019

Abstract

Due to increasing interest in distributed agile software development, there is a need to systematically review the literature on challenges encountered in the agile software development environment. Using the Systematic Literature Review (SLR) approach, 32 relevant publications, dated between 2013 and 2018 were selected from four electronic databases. Data from these publications were extracted to identify the key challenges across the system development life cycle (SDLC) phases, which essentially are short phases in each agile-based iteration. 5 types of key challenges were identified as impacting the SDLC phases; these challenges are Communication, Coordination, Cooperation, Collaboration and Control. In the context of the SLDC phases, the Communication challenge was discussed the most often (79 times, 33%). The least discussed challenges were Cooperation and Collaboration (26 times, 11% each). The 5 challenges occur because of distances which occur in distributed environment. This SLR identified 4 types of distances which contribute to the occurrence of these key challenges - physical, temporal, social-cultural and knowledge/experience. Of the 32 publications, only 4 included research which proposed new solutions to address challenges in agile distributed software development. The authors of this article believe that the findings in this SLR are a resource for future research work to deepen the understanding of and to develop additional solutions to address the challenges in distributed agile software development.

Keywords: Distributed agile development, challenges, systematic literature review, SLR, Scrum, eXtreme Programming (XP), DSDM, FDD, Kanban, Scrumbun, SDLC

1. Introduction

This paper is a Systematic Literature Review (SLR) regarding the challenges of distributed agile development. A SLR is a critical evaluation and summarization of previous research related to the research question(s) [1] and is conducted in a sequence of structured and systematic stages to gather, assess and synthesize the available knowledge [2].

Agile software development methods were originally intended for small projects with co-located team members [3, 4]. Co-location enables face-to-face collaboration between team members which facilitates rapid releases of working software. This results in benefits such as delivering value to the business more frequently than traditional software development methods, enabling more timely changes to the software being developed and improving rates of fixing software defects [5, 6].

Given these benefits, the use of agile was extended to larger-scale software projects, including those where teams work in distributed locations [3, 7]. Using agile for distributed software development has been associated with benefits such as follow-the-sun development which increases overall project productivity, reducing costs by moving work to lower cost countries and having access to a wider pool of talent [5, 7]. However, the use of agile methods in distributed environments is challenging because of the conflict between agility and distribution [25, 29, 32]. The limited or non-existent opportunities for face-to-face interactions is the key root cause of obstacles when using agile software development in a distributed context [5, 7, 8].

After studying several papers from the existing literature, we have identified five challenges in distributed agile software development. We call them “5-Cs of Challenges”. They are control, collaboration, cooperation, coordination, and communication. These challenges are caused by four types of distances. We call them “Distance Factors”. These distances are physical distance, temporal distance, socio-cultural distance and knowledge / experience distance. In order to resolve these challenges, certain researchers have proposed methods. We discuss these methods in this paper as well.

2. Motivation

Given the widespread use of software in companies, and the increasing globalization of businesses, many software development projects are being carried out by distributed teams, and the numbers continue to grow [9]. Many distributed teams use the agile approach for their software development. As per the 2017 report by VersionOne [10], “86% of respondents had at least some distributed teams practicing agile”.

Additionally, 20% of those surveyed in [10] responded that agile helped to improve the management of distributed teams. However, despite the high adoption rates and benefits of distributed agile, there is recognition of the lack of depth of research in this area and the need for more attention from the academic community to understand the impact of agile on distributed teams [3, 6, 54, 55]. Though, some of the areas are still relevant but have been polished significantly such as “trust, and control”. One of the published work [5] was systematic literature review on distributed agile environment but it was a short form of

literature. Thus, it was appropriate to consider relatively a mature era (after 2013), rather than relatively immature era (before 2013) whereas several issues have been resolved already.

3. Research Method

The method used to research this SLR is outlined in Fig. 1 below. This method was developed based on the 3-phase approach described in the technical report by Kitchenham & Charters in [2] and incorporating lessons learnt of conducting SLR within software engineering as described in [11]. Therefore, it was more appropriate to use the enhanced version of [2] in this study.

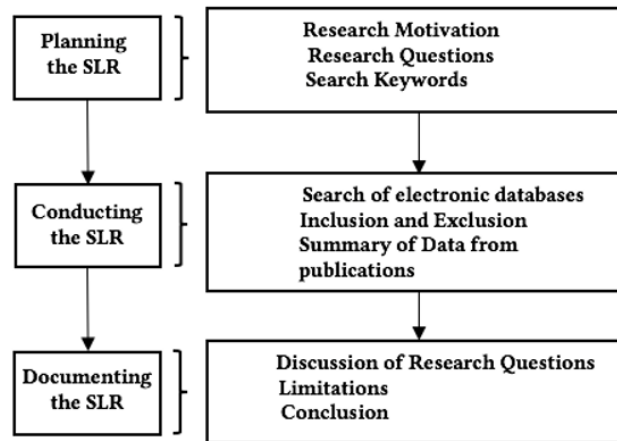


Fig. 1. Research method used in this systematic literature review

3.1 Research Questions

The following questions were studied in this review:

R1: For each of the system development life cycle (SDLC) phases (analysis, design, implementation, testing, deployment and maintenance) within agile iterations, what are the challenges which impact the use of the Agile approach in distributed development teams?

- Motivation for R1: There is a lack of literature that could go deeper into phase-level in agile-based iterations and synthesise challenges scattered in several studies because product owner is sitting somewhere else with the customer. But both are working together in the same iteration.

R2: Based on the challenges identified in R1, what are the most common challenges throughout all the phases?

- Motivation for R2: After analysing challenges in individual phases, it is inadequate if we do not synthesise them together to share an overall picture to the reader, especially in terms of common challenges in each phase during iteration cycles.

R3: For each of the most common challenges identified in R2:

R3-1: What are the factors which contributed to the existence of each of these challenges?

R3-2: What are the recommended solutions to overcome each of these challenges?

- Motivation for R3: There is a need to synthesise factors contributing to the challenges, which are scattered in several studies. Also, it is equally important for the readers to know the available solutions to resolve these challenges.

3.2 Search Keywords

Based on the research questions, set of keywords was constructed to be used in the electronic database search described in section 3.4

- 1) The first part was related to the word challenge and its synonyms; i.e. challenge, issue, problem, obstacle, limitation, difficulty and barrier.
- 2) The second part was “distributed agile”.
- 3) The third part was related to the phases in an Agile development; i.e. analysis, design, implementation, testing, deployment and maintenance.
- 4) The fourth part was related to specific agile methods i.e. Scrum, XP, FDD and DSDM

These search keywords were combined using Boolean operators, resulting in the following search string:

(challenge OR issue OR problem OR obstacle OR limitation OR difficulty OR barrier) AND (distributed AND agile) AND (global AND agile) AND (distributed AND Scrum) AND (distributed AND eXtreme Programming) AND (distributed AND Feature Driven Development) AND (distributed AND Dynamic Systems Development Method) AND (analysis OR design OR implementation OR testing OR deployment OR maintenance)

3.3 Process to Select Set of Publications for Analysis

The phase “Conducting the SLR” as per Fig. 1 above involves the selection of the set of publications for data extraction. The process to identify this set of publications for inclusion in this SLR has 4 steps as illustrated in Fig. 2 below. Details are described in sections 3.4 and 3.5.



Fig. 2. Process to select final set of publications for inclusion

3.4 Search of Electronic Databases

The electronic databases searched (Fig. 2, Step 1) were the ACM digital library, IEEE Xplore, ScienceDirect and SpringerLink. The coverage provided by these 4 databases were deemed sufficiently extensive for this SLR. Within these databases, only English publications in the field of Computer Science. Using the search keywords listed in section 3.2 resulted in a total search result of of 3,695 publications.

3.5 Inclusion and Exclusion Criteria

Using the results of the electronic database search, the next step (Fig. 2, Step 2) in the selection process was a review of titles, keywords, and abstracts against the search strings – those which had words which matched the search strings were included whereas those which did not were excluded. From the 3,695 identified from Step 1, only 59 (2%) were included whereas a large majority (3,636, 98%) of publications were excluded. Apparently, it looks like a drastic reduction in number of papers. The reason we noticed was that, based on these strings, search engines bring up several papers that merely mention “distributed agile” in the text. For instance, this portion of our search string

“(distributed AND Scrum) AND (distributed AND eXtreme Programming) AND (distributed AND Feature Driven Development) AND (distributed AND Dynamic Systems Development Method)”

brings up 87 articles from ScienceDirect, 103 from IEEE Xplore, 109 from SpringerLink and 35 from ACM digital library. Of these almost 334 articles, majority was brought up because they mentioned “distributed” in their titles such as “Reliability and availability issues in large-scale distributed systems”, or “Crowd Sensing Applications: A Distributed Flow-Based Programming Approach”, and many like these. Such papers were absolutely out of the focus of our study hence were not included for further review. This is one of the limitations of search engine as their algorithms are designed to bring up wider range of results which poses difficulty for the researchers of SLR, and waste the time to review irrelevant papers as well.

The search keywords identified in section 3.2 resulted in a broad search string which generated a high number of search results when used in the search engines of the electronic databases being searched. The high number of search results, which required significant manual effort to filter the titles, keywords and abstract to reduce the numbers to a more manageable set for review of the abstracts; had the benefit of reducing selection basis as compared if a more narrow search string had been used.

Table 1 provides the inclusion and exclusion results, by database, from the review of titles, keywords and abstracts.

Table 1. Inclusion and exclusion results from review of titles, keywords and abstracts

Database	Included	Excluded
ACM digital library	5 (5%)	97 (95%)
IEEE Xplore	16 (22%)	56 (78%)
ScienceDirect	10 (1%)	1,365 (99%)
SpringerLink	28 (1%)	2,118 (99%)
Total	59 (2%)	3,636 (98%)

Step 3 in the selection process (Fig. 2, Step 3) was a review of the abstracts of the 59 publications included from the previous step. This review assessed if the abstract provided an indication that the publication contained data relevant to answer the research questions. Publications were included if they did and excluded if otherwise. This resulted in 44 (75%) publications included and 15 (25%) publications excluded and as input into the next step.

The final step (**Fig. 2**, Step 4) in the process was a high level review of the full text of the 44 publications included from the Step 3. This full text high level review assessed if the contents of the publication were relevant to the research questions. A publication was included if the contents were relevant and excluded otherwise. This resulted in 32 (73%) inclusions and 12 (27%) publications being excluded.

4. Types of Distributed Teams

4.1 Distributed Team Members and / or Distributed Teams

Within the 32 publications, 3 different types of distributed projects were discussed. These differences were typically a result of the scope of the research being conducted and not due to different definitions. The 3 types of project distribution described in the publications were:

- 1) Single-team; distributed team members – refers to a single team where team members are distributed in different physical locations: For example [12] described a team of testers which are distributed across 2 European countries and India; and [13] discussed a team where 15 members of a test team are in Norway and China. There were 2 [14, 15] other publications which described this type of distribution ; making it a total of 4 (13%) publications.
- 2) Multiple distributed teams; co-located team members – are multiple teams located in different physical locations; and where team members within the same team are co-located in the same place. For example, [16] described a software that was developed with 16 teams distributed across three continents (Asia, Europe and North America). All team members within a team are co-located to enable face-to-face interactions. [17] discussed the development of a software product where there were 3 development teams – two in different locations in North America and the third team in India – team members in each location were working physically together. 5 other publications [18-22] also researched projects with multiple distributed teams and co-located team members; a total of 7 (22%) publications
- 3) Multiple distributed teams; distributed team members – these are where teams are in different locations and team members are not co-located but also spread out in various places. For example, [23] is a case study of a development project where teams and team members within the teams were distributed across 3 countries (Germany, India and USA). [24] described multiple projects with teams that were distributed across 2 or more countries in Europe. 13 other publications researched projects where both teams and team members were distributed [25-37]; a total of 15 (47%) publications.

Fig. 3 below illustrates these 3 types of distributed and lists the publications in which these types of distribution are discussed.

For the remaining 6 (19%) publications [38-43] the distribution of the projects researched were not specified or unclear

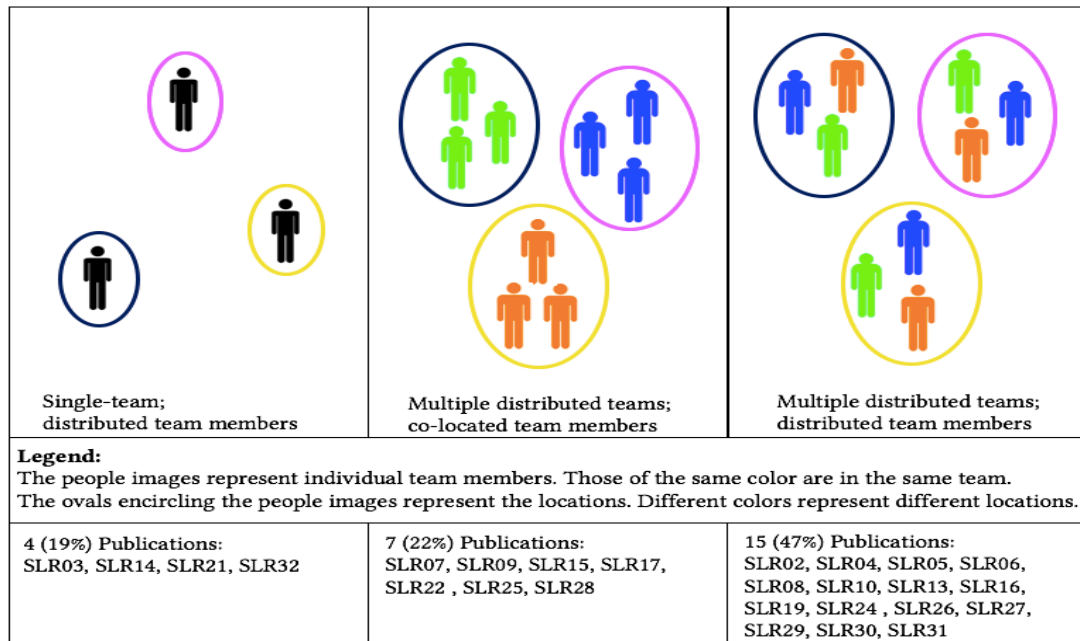


Fig. 3. Types of distributed teams

4.2 Scale of Physical Distribution

The publications included in this SLR described projects which encompassed a range of scale of physical distributions. The least physically distributed were projects which were in the same city but in different locations in the city. The most distributed were projects which were distributed across different countries; with different time zones. **Table 2** provides a summary of the range of scale of distribution described in the publications.

- 1) It was observed that the most limited distribution were projects which were in the same city but in different physical locations [14, 24, 37, 41].
- 2) A slightly larger scale of project distribution was those located in different cities within the same country and with no time difference between the cities [24, 25, 34].
- 3) There were also projects which were distributed across different countries which were in the same time zone [24, 25, 35].
- 4) Most of the publications described projects which had the widest scale of distribution – these projects were distributed across different countries covering different time zones [12-18, 20, 21, 23, 26-31, 36, 38-40, 42].
- 5) There were 5 publications [19, 22, 32, 33, 43], which although there was mention that teams were in different locations, did not specify if these were in the same or different cities or countries.

Table 2. Scale of project distribution

	Scale of Distribution	Publications	Count
Least physically distributed ↑ ↓ Most physically distributed	Distributed across different locations in the same city	[14], [41], [24], [37]	4
	Distributed across different cities in the same country; with no time difference	[25], [24], [34]	3
	Distributed across different countries; with no time difference	[25], [24], [35]	3
	Distributed across different countries; with time zone differences	[38],[25],[12],[26],[27],[28],[18], [29],[19],[36],[39],[40], [23],[14] [16],[30],[17],[31], [13], [20], [42], [21], [26]	22
	Not specific regarding city / country distribution	[19], [43], [32], [33], [22]	5
Total			37 (*)

4.3 Distance Factors Causing Challenges

A key underpinning of the Agile Manifesto [44] is individuals and teams working closely with each other resulting in rapid turnaround of the software being developed. Working closely requires an awareness of the activities which the individuals and teams are working on; and provides context for each other's work [45, 46], thus enabling the close interactions which the manifesto describes. Distributed agile development however introduces distances which create barriers to developing a strong awareness of others. There are 4 key types of distances, which we refer to as the "Distance Factors" affecting distributed agile. The "Distance Factors" are:

- 1) *Physical distance*. This is the most fundamental distance which occurs in distributed projects. This distance refers to individuals or teams who are in different locations and have no face-to-face visibility to each other [14]. We recommend the use of this physical distance instead of geographical distance which is used in some publications. The latter is a more limited definition of distributed (i.e. being in different parts of a city [37], different parts of the same country [24] or different countries [12]. The former is a broader definition of distributed as it encompasses the latter, and includes the additional definition of being in a different part of a floor / building.

- 2) *Temporal distance*. This distance refers to being separated by time zones or working on different shifts with limited or no time overlap [14, 38, 47].
- 3) *Socio-cultural distance*. This is the most complex of distances and refers to individuals / teams where their differences are caused by their socio-cultural environment. These include differences such as different languages, different cultural beliefs, different ethical values and different work practices [47, 48]. Whilst socio-cultural distances exist even with co-located individuals; the physical distance exacerbates the social-cultural distance.
- 4) *Knowledge / Experience distance*. This refers to knowledge / experience that is related to the organization for which the software is being developed for. A project team which is closer to the main hubs of the organization have better knowledge / experience of the organization than those which are further afield [16, 36, 49]. It also refers to the different levels of knowledge / experience across distributed teams [20].

These “Distance Factors”, which occur when work is distributed, contribute to the existence of challenges with the agile approach which is premised on proximity. Based on the data from the publications reviewed for this SLR, we propose grouping these challenges into 5 categories, which we refer to as the “5-Cs of Challenges” - Communication, Coordination, Cooperation, Collaboration and Control. The “5-Cs of Challenges” is developed based on previous frameworks to categorize challenges in software development [47, 50]. The uniqueness of the proposed “5-Cs” grouping is that whilst previous literature described subsets of the five challenges, our research has identified that assessment of the challenges of distributed agile software development should be done from all 5 areas to provide a more comprehensive perspective. Additionally, it is the addition of the understanding of the “Distance factors” which contribute to the existence of the “5-Cs” that adds to the uniqueness of our findings. It is worth noting that whilst Cooperation and Collaboration are sometimes used interchangeably, these 2 terms have subtly different meanings - in particular, Collaboration requires Cooperation whereas Cooperation can exist without Collaboration [53].

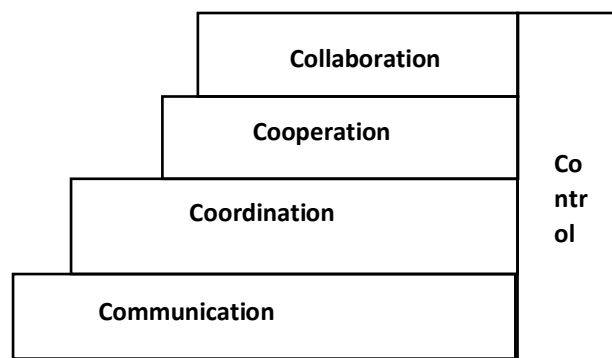


Fig. 4. The “5-Cs of challenges”

The “5-Cs of Challenges” is depicted visually in **Fig. 4** and described in more detail below.

- Communications refers to unambiguous and transparent sharing of information - formally and / or informally; verbally and / or written; in hardcopy and / or digital

format [47, 50]. Project documentation, knowledge management systems and training are considered part of communications.

- Coordination is the sequencing and integration of inter-dependent activities to achieve a common objective / outcome [47, 50].
- Cooperation is when two (or more) individuals / teams have mutually agreed to work on activities which have been divided amongst each other [52, 53].
- Collaboration takes cooperation to the next level - where individuals / teams not only agree to work together but do so for mutual benefit and develop deep meaningful relationships when working together [52-53].
- Control refers to the governance and project management activities in system development [47, 51]. Provision of the required work environment (e.g. office space, hardware and software) fall in this category.

We thus propose the “Distance Factors and 5-Cs of Challenges” framework as discussed in this SLR.

5. Discussion of Research Questions

5.1 Answer to R1- Challenges by Phases

Before we elaborate these challenges, it is useful to mention that there are several obvious differences in agile and traditional development approaches. Majority of them is related to the “practices” in a development environment such as customer involvement, less design, more collaboration among team members. However, in terms of common features, one of the fundamental commonalities between waterfall method and agile approach is that agile approaches e.g., Scrum, FDD, DSDM, Kanban, Scrumban are phase-based methods, like we have in waterfall. In these methods, team literally have to execute waterfall like phases (user stories/requirements analysis, design, implementation, testing, deployment and maintenance). But these are shorter phases in agile environment as compared to a traditional waterfall method. Agile approaches e.g., Lean and XP are practice-based approaches (no-phases in them so far). These practice-based approaches work inside these phases such as XP preaches to have less documentation or design in design phase and so on. The purpose of short – phases is to achieve frequent delivery, early feedback and/or fail-fast among others.

5.2 Answer to R2 – The Most Common Challenges

Based on Fig. 5 which is visualization of the frequency distribution of the number of times the 5-Cs are discussed in the publications in this SLR, the most common challenge is Communications (at 33%) followed by Coordination (28%). The other 3 challenges are discussed significantly less frequently; i.e. $\geq 10\%$ less frequent than the Communication and Coordination challenges.

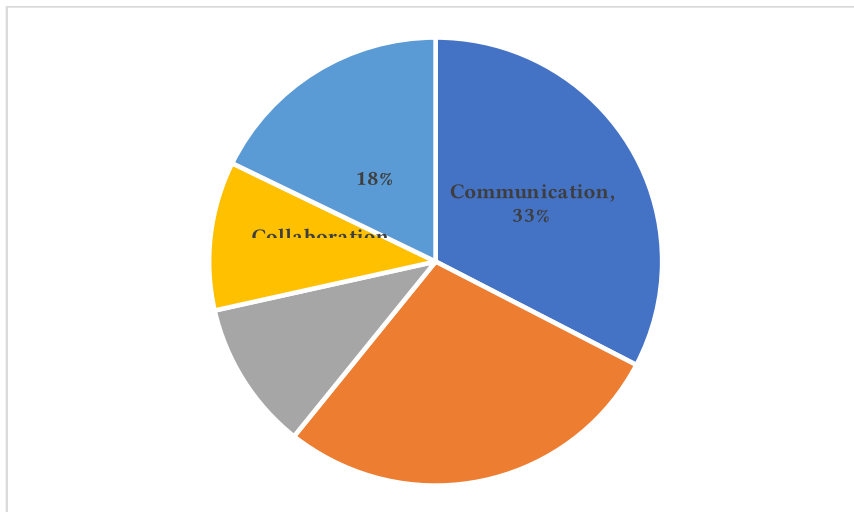


Fig. 5. Frequency of discussion of the 5-Cs

Within the communication category of challenges for agile distributed projects, there exists the challenge between onshore and offshore software development. An offshore site team requires efficient communication to reduce the iteration time with teams at the onshore sites. However, it is not only onshore and offshore teams which face communication challenges. Customers in a distributed environment also face difficulties in resolving project issues due to the limited opportunity for face-to-face communications.

5.3 Answer to R3-1 – Factors Contributing to the 2 Most Common Challenges (Communication and Coordination)

For the 2 most common challenges (Communication and Coordination) as identified in section 5.1, the frequency of the distances which contribute to the occurrence of each of these challenges, as discussed in the SLR publications, are summarized in **Table 3** below.

Table 3. Distances which contribute to the communication and coordinate challenges

	Physical	Temporal	Socio-Cultural	Knowledge / Experience
Communication	29 (40%)	23 (32%)	17 (24%)	3 (4%)
Coordination	25 (35%)	20 (28%)	16 (22%)	3 (4%)

Fig. 6 below is a visual representation of the frequencies of the distances. As these the figure shows, the most frequently occurring distance which affects both Communications and Coordination challenge is the physical distance. This is to be expected given that, physical distance is the most fundamental distance which occurs in distributed projects. Temporal

distance is the second most described distance – this distance only occurs for projects where teams are distributed in different time zones or are working in different shifts with no time overlap. Socio-cultural distances (languages, cultural beliefs, ethical values and work practices) which is described the third most frequently, may occur despite being co-located; but in the case of distributed environments, the publications describe the physical distance as exacerbating the socio-cultural distance, thus leading to communication and coordination challenges. Knowledge / Experience distances is the least researched of the distances which cause challenges in agile distributed software development – and thus is described the least frequent.

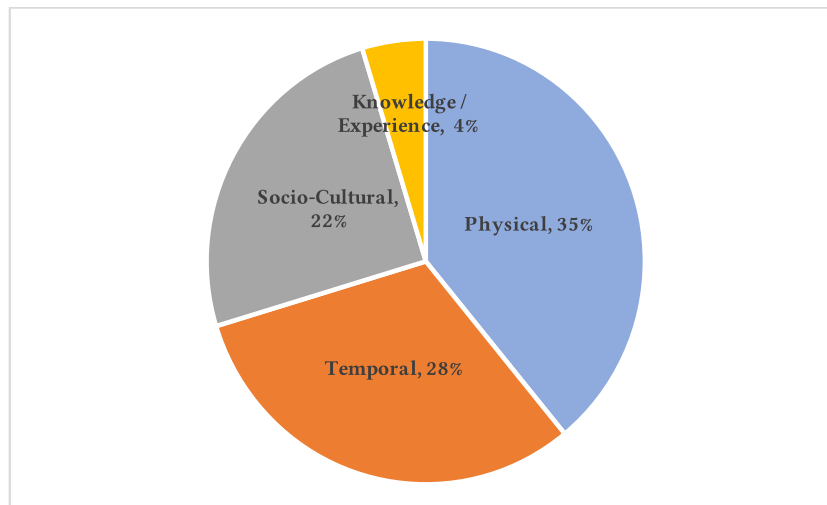


Fig. 6. Distances contributing to the coordination challenge

5.4 Answer to R3-2 – Solutions to Overcome the 2 Most Common Challenges (Communication and Coordination)

We identified 4 publications [17, 29, 33, 41] where new methods were developed and tested as part of the research. 2 publications proposed methods to specific address the most common challenge – i.e. the Communications challenge. The other 2 methods developed were applicable to all the 5-Cs, and thus would also be relevant to overcome the second most common challenge (Coordination). The remaining 28 publications did not identify new methods to overcome the 5-Cs – they either provided lessons learnt based on projects researched or did not provide any solutions.

- 1) Solutions to overcome Communication Challenge:
SLR17 proposed that distributed communications could be improved through a structured approach to waste identification and mitigation [17]. SLR18 developed a “feature tree” [41] to improve the communications of changes in requirements throughout the software development process.
- 2) Solutions applicable to overcome all 5-Cs:
The proposed frameworks in SLR08 and SLR27 were targeted at helping to address all 5 of the challenges. SLR08 described the “PBRUC patterns-based unsupervised

requirements clustering framework” [29] for validating requirements between distributed stakeholders. In SLR27, a framework to manage risks was proposed based on inputs from the industry [33].

6. Conclusion

In this SLR we identified the key challenges across the agile-based iterations including short SDLC phases within iterations. Five types of key challenges were identified as impacting the SDLC phases within agile-based iterations. We called these the “5-Cs of Challenges” : Communication, Coordination, Cooperation, Collaboration and Control. We also identified 4 types of distances which contributed to the occurrence of these 5 types of key challenges. These distances are physical, temporal, social-cultural and knowledge/experience. Based on these, we proposed the “Distance Factors and 5-Cs of Challenges” framework to categorize the findings of this SLR. The framework also provides a guide as to the category of challenge that could occur if a type of distance factor was present in a agile software development project. We also identified that there were limited new proposed solutions to help address challenges in distributed agile environment - of the 32 selected publications, only 4 proposed new solutions to address challenges in agile distributed software development. Given that this research was focused on challenges faced by the distributed agile approach at large, and excluded specific agile methods, models, practices and frameworks, there is opportunity for future research to assess if there are challenges caused by use of specific tools.

References

- [1] A. Schwarz, M. Mehta, N. Johnson, and W. W. Chin, "Understanding frameworks and reviews: a commentary to assist us in moving our field forward by analyzing our past," *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, vol. 38, no. 3, pp. 29-50, 2007. [Article \(CrossRef Link\)](#)
- [2] B. Kitchenham, and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," *Technical report, Ver. 2.3 EBSE Technical Report. EBSE. Keele University*, 2007.
- [3] P. Abrahamsson, O. Salo, J. Ronkainen, and J. Warsta, "Agile software development methods: Review and analysis," *arXiv preprint arXiv: 1709. 08439*, 2017.
- [4] B. Boehm and R. Turner, "Management challenges to implementing agile processes in traditional development organizations," *IEEE software*, vol. 22, no. 5, pp. 30-39, 2005. [Article \(CrossRef Link\)](#)
- [5] E. Hossain, M. A. Babar, and H.-y. Paik, "Using scrum in global software development: a systematic literature review," in *Proc. of Global Software Engineering, 2009. ICGSE 2009. Fourth IEEE International Conference on*, pp. 175-184, 2009. [Article \(CrossRef Link\)](#)
- [6] A. M. Razavi and R. Ahmad, "Agile development in large and distributed environments: A systematic literature review on organizational, managerial and cultural aspects," in *Proc. of Software Engineering Conference (MySEC), 2014 8th Malaysian*, pp. 216-221, 2014. [Article \(CrossRef Link\)](#)
- [7] F. Lanubile, C. Ebert, R. Prikladnicki, and A. Vizcaíno, "Collaboration tools for global software engineering," *IEEE software*, vol. 27, no. 2, pp. 52-55, 2010. [Article \(CrossRef Link\)](#)
- [8] P. Abrahamsson, K. Conboy, and X. Wang, "'Lots done, more to do': the current state of agile systems development research," *European Journal of Information Systems*, vol. 18, no. 4, pp. 281-284, 2009. [Article \(CrossRef Link\)](#)

- [9] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE software*, vol. 18, no. 2, pp. 16-20, 2001. [Article \(CrossRef Link\)](#)
- [10] V. One, "Releases 11th Annual State of Agile™ Report," *Technical report*, 2017. [Article \(CrossRef Link\)](#)
- [11] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80, no. 4, pp. 571-583, 2007. [Article \(CrossRef Link\)](#)
- [12] T. Anand and V. Mani, "Practices to make agile test teams effective: challenges and solutions," in *Proc. of 2015 IEEE 10th International Conference on Global Software Engineering Workshops (ICGSEW)*, pp. 7-11, 2015. [Article \(CrossRef Link\)](#)
- [13] N. B. Moe, D. Cruzes, T. Dybå, and E. Mikkelsen, "Continuous software testing in a globally distributed project," in *Proc. of Global Software Engineering (ICGSE), 2015 IEEE 10th International Conference on*, pp. 130-134, 2015. [Article \(CrossRef Link\)](#)
- [14] A. M. E. Hamid, "Upgrading distributed agile development," in *Proc. of presented at the 2013 International Conference on Computing, Electrical and Electronic Engineering (ICCEEE)*, 2013. [Article \(CrossRef Link\)](#)
- [15] A. Yagüe, J. Garbajosa, J. Díaz, and E. González, "An exploratory study in communication in Agile Global Software Development," *Computer Standards & Interfaces*, vol. 48, pp. 184-197, 2016. [Article \(CrossRef Link\)](#)
- [16] M. M. Jha, R. M. F. Vilardell, and J. Narayan, "Scaling Agile Scrum Software Development: Providing Agility and Quality to Platform Development by Reducing Time to Market," in *Proc. of Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on*, pp. 84-88, 2016. [Article \(CrossRef Link\)](#)
- [17] M. Korkala and F. Maurer, "Waste identification as the means for improving communication in globally distributed agile software development," *Journal of Systems and Software*, vol. 95, pp. 122-140, 2014. [Article \(CrossRef Link\)](#)
- [18] J. M. Bass, "Artefacts and agile method tailoring in large-scale offshore software development programmes," *Information and Software Technology*, vol. 75, pp. 1-16, 2016. [Article \(CrossRef Link\)](#)
- [19] R. Bin-Hezam and S. Alyahya, "Managing customer involvement in globally distributed agile projects," in *Proc. of Global Software Engineering Workshops (ICGSEW), 2016 IEEE 11th International Conference on*, pp. 7-12, 2016. [Article \(CrossRef Link\)](#)
- [20] N. B. Moe, D. Šmite, A. Šablīs, A.-L. Börjesson, and P. Andréasson, "Networking in a large-scale distributed agile project," in *Proc. of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, p. 12, 2014. [Article \(CrossRef Link\)](#)
- [21] A. Sablis and D. Smite, "Agile Teams in Large-Scale Distributed Context: Isolated or Connected?," in *Proc. of the Scientific Workshop Proceedings of XP2016*, p. 10, 2016. [Article \(CrossRef Link\)](#)
- [22] E. Thomson and R. Vidgen, "Balancing the Paradox of Formal and Social Governance in Distributed Agile Development Projects," *Information Systems Development: Springer*, pp. 155-166, 2013. [Article \(CrossRef Link\)](#)
- [23] R. K. Gupta, P. Manikreddy, and K. Arya, "Pragmatic Scrum Transformation: Challenges, Practices & Impacts During the Journey A case study in a multi-location legacy software product development team," in *Proc. of the 10th Innovations in Software Engineering Conference*, pp. 147-156, 2017. [Article \(CrossRef Link\)](#)
- [24] I. Richter, F. Raith, and M. Weber, "Problems in agile global software engineering projects especially within traditionally organised corporations:[An exploratory semi-structured interview study]," in *Proc. of the Ninth International C* Conference on Computer Science & Software Engineering*, pp. 33-43, 2016. [Article \(CrossRef Link\)](#)
- [25] Y. I. Alzoubi, A. Q. Gill, and A. Al-Ani, "Empirical studies of geographically distributed agile development communication challenges: A systematic review," *Information & Management*, vol. 53, no. 1, pp. 22-37, 2016. [Article \(CrossRef Link\)](#)

- [26] D. Badampudi, S. A. Fricker, and A. M. Moreno, "Perspectives on Productivity and Delays in Large-Scale Agile Projects," in *Proc. of International Conference on Agile Software Development*, pp. 180-194, 2013. [Article \(CrossRef Link\)](#)
- [27] A. Banijamali, M. O. Ahmad, J. Similä, M. Oivo, and K. Liukkunen, "Empirical Investigation of Scrumban in Global Software Development," in *Proc. of International Conference on Model-Driven Engineering and Software Development*, pp. 229-248, 2016.
- [28] J. M. Bass, "How product owner teams scale agile methods to large distributed enterprises," *Empirical Software Engineering*, vol. 20, no. 6, pp. 1525-1557, 2015. [Article \(CrossRef Link\)](#)
- [29] P. Belsis, A. Koutoumanos, and C. Sgouropoulou, "PBURC: a patterns-based, unsupervised requirements clustering framework for distributed agile software development," *Requirements engineering*, vol. 19, no. 2, pp. 213-225, 2014. [Article \(CrossRef Link\)](#)
- [30] Y. Khmelevsky, X. Li, and S. Madnick, "Software development using agile and scrum in distributed teams," in *Proc. of Systems Conference (SysCon), 2017 Annual IEEE International*, pp. 1-4, 2017. [Article \(CrossRef Link\)](#)
- [31] P. Lous, M. Kuhrmann, and P. Tell, "Is Scrum fit for global software engineering?," in *Proc. of the 12th International Conference on Global Software Engineering*, pp. 1-10, 2017. [Article \(CrossRef Link\)](#)
- [32] S. V. Shrivastava and U. Rathod, "Categorization of risk factors for distributed agile projects," *Information and Software Technology*, vol. 58, pp. 373-387, 2015. [Article \(CrossRef Link\)](#)
- [33] S. V. Shrivastava and U. Rathod, "A risk management framework for distributed agile projects," *Information and software technology*, vol. 85, pp. 1-15, 2017. [Article \(CrossRef Link\)](#)
- [34] R. Vallon, S. Strobl, M. Bernhart, and T. Grechenig, "Inter-organizational co-development with scrum: experiences and lessons learned from a distributed corporate development environment," in *Proc. of International Conference on Agile Software Development*, pp. 150-164, 2013. [Article \(CrossRef Link\)](#)
- [35] V. J. Wawryk, C. Krenn, and T. Dietinger, "Scaling a running agile fix-bid project with near shoring: Theory vs. reality and (best) practice," in *Proc. of Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on*, pp. 1-7, 2015. [Article \(CrossRef Link\)](#)
- [36] D. S. Cruzes, N. B. Moe, and T. Dybå, "Communication between developers and testers in distributed continuous agile testing," in *Proc. of Global Software Engineering (ICGSE), 2016 IEEE 11th International Conference on*, pp. 59-68, 2016. [Article \(CrossRef Link\)](#)
- [37] R. Vallon, C. Dräger, A. Zapletal, and T. Grechenig, "Adapting to Changes in a Project's DNA: A Descriptive Case Study on the Effects of Transforming Agile Single-Site to Distributed Software Development," in *Proc. of presented at the 2014 Agile Conference, Kissimmee, FL, USA, 2014*. [Article \(CrossRef Link\)](#)
- [38] L. H. Almeida and A. B. Albuquerque, "A multi-criteria model for planning and fine-tuning distributed scrum projects," in *Proc. of Global Software Engineering (ICGSE), 2011 6th IEEE International Conference on*, pp. 75-83, 2011. [Article \(CrossRef Link\)](#)
- [39] K. Dikert, M. Paasivaara, and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp. 87-108, 2016. [Article \(CrossRef Link\)](#)
- [40] S. Fellhofer, A. Harzl, and W. Slany, "Scaling and Internationalizing an Agile FOSS Project: Lessons Learned," in *Proc. of IFIP International Conference on Open Source Systems*, pp. 13-22, 2015. [Article \(CrossRef Link\)](#)
- [41] D. Lloyd, R. Moawad, and M. Kadry, "A supporting tool for requirements change management in distributed agile development," *Future Computing and Informatics Journal*, vol. 2, no. 1, pp. 1-9, 2017. [Article \(CrossRef Link\)](#)
- [42] M. A. Razzak, T. Bhuiyan, and R. Ahmed, "Knowledge management in distributed agile software development projects," in *Proc. of IFIP International Workshop on Artificial Intelligence for Knowledge Management*, pp. 107-131, 2014. [Article \(CrossRef Link\)](#)

- [43] S. McGinnes, "Barriers to client collaboration in agile offshore Information Systems development," *Information Systems Development: Springer*, pp. 601-612, 2013. [Article \(CrossRef Link\)](#)
- [44] K. Beck et al., "Manifesto for Agile Software Development," 2009. [Article \(CrossRef Link\)](#)
- [45] P. Dourish and V. Bellotti, "Awareness and coordination in shared workspaces," in *Proc. of the 1992 ACM conference on Computer-supported cooperative work*, pp. 107-114, 1992. [Article \(CrossRef Link\)](#)
- [46] D. Damian, L. Izquierdo, J. Singer, and I. Kwan, "Awareness in the wild: Why communication breakdowns occur," in *Proc. of Global Software Engineering, 2007. ICGSE 2007. Second IEEE International Conference on*, pp. 81-90, 2007. [Article \(CrossRef Link\)](#)
- [47] P. J. Agerfalk, B. Fitzgerald, H. Holmstrom Olsson, B. Lings, B. Lundell, and E. Ó Conchúir, "A framework for considering opportunities and threats in distributed software development," 2005.
- [48] H. Holmstrom, E. Ó. Conchúir, J. Agerfalk, and B. Fitzgerald, "Global software development challenges: A case study on temporal, geographical and socio-cultural distance," in *Proc. of Global Software Engineering, 2006. ICGSE'06. International Conference on*, pp. 3-11, 2006. [Article \(CrossRef Link\)](#)
- [49] S. Modi, P. Abbott, and S. Counsell, "Exploring communication challenges associated with Agile practices in a globally distributed environment," in *Proc. of RAISE 2012-Researching Agile Development of Information SystEms Conference*, 2012.
- [50] C. A. Ellis, S. J. Gibbs, and G. Rein, "Groupware: some issues and experiences," *Communications of the ACM*, vol. 34, no. 1, pp. 39-58, 1991. [Article \(CrossRef Link\)](#)
- [51] E. Carmel and R. Agarwal, "Tactical approaches for alleviating distance in global software development," *IEEE software*, vol. 18, no.2, pp. 22-29, 2001. [Article \(CrossRef Link\)](#)
- [52] H. Fuks, A. Raposo, M. A. Gerosa, M. Pimentel, D. Filippo, and C. Lucena, "Inter-and intra-relationships between communication coordination and cooperation in the scope of the 3C Collaboration Model," in *Proc. of Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on*, pp. 148-153, 2008. [Article \(CrossRef Link\)](#)
- [53] R. Ashkenas, "There's a difference between cooperation and collaboration," *Harvard Business Review*, vol. 20, 2015.
- [54] B. Ramesh, L. Cao, K. Mohan, and P. Xu, "Can distributed software development be agile?," *Communications of the ACM*, vol. 49, no. 10, pp. 41-46, 2006. [Article \(CrossRef Link\)](#)
- [55] R. Vallon, B. J. da Silva Estácio, R. Prikładnicki, and T. Grechenig, "Systematic literature review on agile practices in global software development," *Information and Software Technology*, vol. 96, pp. 161-180, 2018. [Article \(CrossRef Link\)](#)



Imran Ghani is an Associate Professor at Mathematical and Computer Sciences Department, Indiana University of Pennsylvania, United States. He received his Master of Information Technology Degree from UAAR (Pakistan), M.Sc. Computer Science from UTM (Malaysia) and Ph.D. from Kookmin University (South Korea). His research focus includes agile software development practices, semantics techniques, web services, software testing.



Angelica Lim is a business investor. She who was previously the Chief Financial Officer at BP Petrochemical, Head of BP Business Service Centre Asia and a partner at Accenture. She has a Bachelor of Science (Honours) and Masters of Business Administration; both from the University of Melbourne. She is currently applying to do PhD Research in Big Data Analytics.



Mr. Muhammad Hasnain is currently a post graduate student of Monash University Malaysia Campus. He received his Master of Computer Science (MCS) degree from (UAAR) and Master of Science in Computer Science (MSCS) degree from Abasyn University Pakistan. His research focus includes on regression testing, fuzzy logic, machine learning, web services performance testing, and web services selection and ranking.



Israr Ghani is a PhD student at School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia Johor Bahru. He received his Master of Computer Science Degree from UAAR (Pakistan), His research focus includes software system analysis, web services, web services testing, and software testing.



Dr. Muhammad Imran Babar received his Ph.D. degree in Computer Science from Universiti Teknologi Malaysia (UTM) in 2015. Currently, he is serving as an Assistant Professor at Army Public College of Management & Sciences (APCOMS) Rawalpindi, Pakistan. He is Head of Department in Department of Computer Science & Software Engineering at APCOMS. His research interests are in software engineering, decision support systems, artificial intelligence, neural networks, predictive analytics, fuzzy logic and big data.