AGILE IN DISTRIBUTED SOFTWARE DEVELOPMENT: A SYSTEMATIC LITERATURE REVIEW

Evis TRANDAFILI, Besjana MURAKU and Elinda MEÇE
Department of Computer Engineering, Faculty of Information Technology, Polytechnic University of Tirana, Albania

ABSTRACT

Distributed teams have long adapted their own versions of implementation the of Agile methods, and the use of many tools to facilitate the access to these methods. However, which is the most appropriate method and the factors impacting it remain unclear. A systematic literature review of the most recent publications to identify the challenges, trends, and the less explored area of Agile Methodologies for distributed software development (DSD) is here made for the years 2017-2021 by including here the pandemic years. The results showed a considerable interest in Agile for Distributed Software Development. Case studies and guidelines and proposal of hybrid models' implementation were considered for the review. Scrum and Extreme Programming are the most reviewed methods with focus on Peer Programming and Lean Programming. At the same time, an important attention is given to scaled agile and its adoption for distributed teams too, where most of the analysis is focused on Scrum of Scrums, Scaled Agile Framework and Disciplined Agile Delivery.

Keywords: Agile, global software development, distributed software development, scurm

1. INTRODUCTION

Global software development (GSD) has an increasing interest both in industry and in academics. The decision to go global for many companies is considered cost-effective, time reducing thus faster development, access to global resources and increased flexibility. In addition, it faces several challenges such as geographical locations, cultures, time zones and languages all affected the team involved, the project structure and processes.

GSD-related topics have been in focus and discussed extensively in literature. However, understanding and examining all the aspects of GSD is particularly challenging as it varies by the organization type and structure (how are teams are organized and distributed), the form of GSD implementation (whether the teams are independent or distributed), the team experience. It is noticed that each publication usually focuses on a particular challenge and project aspect.

Agile

Unlike the traditional software process development methods that are plan-driven and have more strict phases, agile methods have introduced a dynamic process with less documentation produced but more software delivered in less time, thus responding better to business requirements. In such a powerful procedure, prerequisites are profoundly unpredictable and consistent joint effort is basic to adapt to constantly changing necessities for chance alleviation because of conditions (Martakis and Deneva 2013). Engineer coordinated effort is subject to the correspondence of changes of new errands, just as on the consciousness of what others are doing and whether they are accessible to help (Damian *et al*., 2007).

Agile in distributed teams

There are several publications about Agile and global software development. In recent publications, authors have tried to give best practices also propose new models to minimize the challenges faced. However, there is a lack of information from real cases of industry which results could be found in (Vallon 2018).

Consequently, we will investigate the recent case studies or evidences coming from the industry and identify the trends in this area aiming to provide information about the evidences provided, factors impacting agile development in distributed software development, whether there are new models proposed or novel approaches for Agile in DSD, and the most studied methods in distributed Agile.

2. RESEARCH RESULTS

Information has been obtained from databases of ACM Digital Library, AIS Electronic Library (AISeL), IEEE Xplore, ScienceDirect and any open-source publication. The research query is composed of two main objectives: Agile and global software development. Taking into consideration that there may be different combinations and sub queries, the last version of the query is as follows:

(Agile OR scrum OR "extreme programming" OR "pair programming" OR "lean programming" OR dsdm OR kanban) AND ("global software development" OR GSD OR "virtual team" OR "global team") AND ("empirical study" OR "case study").

The systematic literature review was implemented in two phases. First, all the papers resulting from the query were reviewed and independently selected by two authors. Second, the papers were categorized and reviewed.

There were 100 articles only for the period of 2020-2021. Once the manuscripts' content was read only 14 papers were selected, relevant to our systematic review. There were 46 papers selected in total for this review.

Following, we provide answers to the research questions by giving in depth details and explanation.

2.1. What evidence are provided (surveys/ case studies)?

The manuscripts could be divided into five categories. Three articles are included in the group of proposal papers for future research (could also considered as 'work in progress') where the authors proposed the field of study and the methods that will be using. Five papers are classified as 'Surveys'. Sixteen publications are classified as 'case studies,' where authors have provided insights from personal experience or real projects of Agile in DSD. There are also identified two papers included into the group of the 'New Model Proposal' papers.

Table 7 Papers categorisation. Total findings for each category.

Category	Papers	Total
Proposal for research	(Razzak, 2017), (Lunesu *et al.*, 2018), (Drechsler and Breth, 2019)	3
Surveys	(Boyer and Mili, 2011), (Werewka *et al.*, 2017), (Seckin *et al.*, 2018), (Vithana *et al.*, 2018), (Marinho, 2019), (Majdenbaum and Chaves, 2020) (Shameem *et al*, 2020)	7
Case Studies	(Awar *et al.*, 2017), (Inayat *et al.* 2017), (Santos and Nunes, 2017), (Bass *et al.*, 2018), (Costa *et al.*, 2018), (Kahya and Seneler, 2018 (a)), (Kahya and Seneler, 2018 (b)), (Rajpal, 2018), (Paramartha, 2018), (Aggarwal and Mani, 2019), (Bjørn *et al.*, 2019) (Gupta *et al.*, 2019 (a)), (Gupta *et al.*, 2019(b)), (Salameh and Bass, 2019), (Szabó and Steghöfer, 2019), (Uludağ *et al.*, 2019), (Qahtani, 2020), (Shafiq *et al.*, 2020), (Moe *et al.*, 2020), (Britto *et al.*, 2020), (McCarthy *et al.*, 2020), (Stray and Moe, 2020), (Khan *et al.*, 2021), (Beecham *et al.*, 2021), (Geeling *et al.*, 2020)	25
Findings from other case studies	(Khmelevsky *et al.*, 2017), (Lous *et al.*,2017), (Humble, 2018), (Putta, 2018), (Calefato *et al.*,2020), (Camara *et al.*, 2020), (Shafiq *et al.*, 2020)	7

New models proposals	(Awar *et al.*, 2017), (Kroll *et al.*,2017), (Beecham *et al.*, 2021)	3

2.2. What factors impact Agile in Distributed Software Development?

Here the challenges and the drawbacks of Agile in global software development are identified along with the positive aspects that these methodologies combined with distributed software development might have brought.

Some of the main characteristics that global software development needs to address are related to temporal, geographical and sociocultural distances that pose challenges in communication, control, and coordination (Szabó and Steghöfer, 2019).

Agile practices require frequent communications in the team. In the research conducted by (Inayat et al. 2017) are investigated multiple-case study of four large, distributed companies. They found that in some companies, where the teams were involved in distributed projects, communication with the local colleagues was more frequent than with the remote ones and as a result, the members know less about the professional background and are less aware of the tasks that the remote colleague was working on. The work progress of the remote team was also not transparent. This resulted in people being more likely to communicate with someone they knew and that they knew they can help, thus communicating more with those locally than remotely. However, their study, despite the apparent believe, indicates that distance does not seem to matter to communication frequency, and the correlation results between communication, awareness and distance are indecisive. Further investigations need to be conducted for wider cases studies.

A particular case of GSD, where programming is appropriated over a twenty-four-hour working day, is Follow the Sun (FTS) (Carmel *et al.*, 2014). For projects like this, the lack of communication or interaction in real time, the time difference due to no overlapping of working hours or delays in response times for problems, were key challenges for the temporal distance (Kahya and Seneler, 2018). Also, the loss of concentration because of long meetings at late hours is another risk.

As suggested by (Khmelevsky *et al.*, 2017) the lack of face-to-face communication, because of the distribution of the team, should be compensated with rich communications using channels and as in real agile, at least weekly, or biweekly meetings between the teams should be hold. Brainstorming and frequent planed meetings are essential to overcome problems for distributed team.

In Agile there are specific roles that conduct specific activities. The Product Owner, that has knowledge about the system being developed, should be close to the developing team to interact with them (Kelly and Allan, 2019). In cases where the product owner is the client itself, in distributed environment it might not be possible. Thus, for distributed software development projects, it is recommended that the Product Owner should be a member of the team that has the feasibility to relocate near the customer to discuss the business requirements and translate them in user stories for the developing team (Paramartha, 2018). The role of the Product Owner may be found also with the name of Business Analyst. In case of frequent traveling and the utilization of tools for communication and implementing requests an adequate budged to be estimated (Rajpal, 2018).

On research conducted for DSD in India and US or Europe, is observed that the IT developers of India work under various transactional conditions that differ from those who work in US or Europe (Bjørn *et al.,* 2019). The idea of trans locality guides us to consider the accomplished work arrangements as a variety of word related governmental issues, infrastructural availability, and worldwide office, which reach past national fringes. By focusing on the manners by which procedure, work organization and technology shape the trans locality of the working environment we can thus understand their lived work experience in transnational work. Agile methodologies have an advantage of increasing transparency and coordination across team; however, this can be a risk for vendor companies offshore since it can disempower the developers by reducing their decisions, thus having a negative impact in their work (Bjørn *et al.*, 2019).

Technology diversity is another factor impacting distributed software development. Agile methodologies require tools for team communication and coordination. Sometimes, different teams work with different tools and the time of the adaptation is requested. In other occasions the remote team may pose resistance in moving from their own internal environment to something new.

Awar *et al.*, (2017) identified the English language as the one of the most challenging issues. Different level of comprehension of the communication language led to misunderstanding or difficulties in understanding requirements or tasks assigned to the development team.

Using the Hofstede model, Lunesu *et al.*, (2018) rises hypotheses with respect to effect of social foundation on rehearses appropriation of the teams involved. Ethnic, social, and cultural aspects determine the diversity in socio-cultural. Even though they are still hypothesis, whenever approved, would assist light-footed professionals with identifying early the potential difficulties that they will confront unavoidably (and in this manner to be increasingly arranged to this challenges)

Other challenges are related to correspondence, specialized competency, client commitment, information exchange, innovation, coordination, and control. However, training and coaching and community-oriented improvement were not found to have a noteworthy relationship with project success for distributed teams (Vithana *et al.*, 2018). They are close to personal attitude and building the consistent project success.

One of the most important aspects for organization is lowering development cost, and this can be achieved by going global and delegating the work in other countries (Humble, 2018). In Global Software Engineering (GSE), there are identified three success factors (competences, communication, and collaboration) and three benefits (flexibility, innovation, and efficiency) (Elbert *et al.*, 2016).

Agile methods are developed around communication and transparency that they imply and have a positive impact in global teams. These methods encourage frequent meetings and collaborations that help to reduce the gap between development teams. There are several tools that are used for collaboration (Jira, Cacoo, GenMyModel), for shared project workspace (like GitHub or Subversion) and communication in the team (Skype, GoogleHangouts) (Calefato and Ebert., 2019). Stray and Moe (2020) investigated the impact of instant messaging tool, Slack, in reducing the challenges of geographic distance. This tool supports frequent communication and fast responses within and between teams and their stakeholders, which in turn benefit GSE companies. However, even in mature agile GSE companies using new tools and coordinating with both scheduled and unscheduled meetings, faces the same old barriers – such as language, unbalanced activity, and difficulty with facilitating communication, the authors say.

2.3. Are there new models proposed or novel approaches for Agile in DSD?

Task scheduling is challenging to Global Software Development (GSD). Usually this is a process carried by the project managers who are responsible of creation and distribution of tasks. The process or task assignment becomes especially difficult considering the distribution, the multidisciplinary composition of the team and time zone differences. For FTS projects, the time zone difference is exploited as an advantage of GSD, however poor planning and a poor distribution of the tasks can increase the costs in an unacceptable manner (Penta *et al*., 2011).

Kroll *et al.*, (2017) have implemented a genetic algorithm-based assignment technique that uses a queue-based GSD simulator for fitness function evaluation. Their work contributes on task scheduling in conform to GSD context. Genetic Algorithm-based (GA) have been widely used in many optimizations, search, and machine learning (ML) problems (Camara *et al.*, 2020). This algorithm was also evaluated for task scheduling in three industrial project's data, and then the results have been confronted with the actual PMs (project managers).

Kroll *et al*., (2017) stated that in terms of reducing the project lifespan, the Genetic Algorithm performed as better as, and in cases even better than solution provided by the managers for task assignment to the development team.

Awar *et al*., (2017) proposed new model which is based on practices and state-of –the-art for the process of software development in distributed Agile team. This model divides the process into four phases: pre-implementation, where the goal here is to create a set a baseline for the cross-functional team; implementation, where the team should implement fully the Agile methodology; team-shared understanding, is the phase where issues are looked up by appropriated group. The last phase, post-Development, is expected to fortify the association by utilizing certain practices. However, this model is applied so far only in one case and further investigations need to be considered.

Sinha *et al*., (2020) suggested another model called SWOT model (strengths, weaknesses, opportunities, threats). What is important about this model is that it highlights some of the most principal factors which may influence the organization's future in a GSD system. The findings of this study reported 24 factors among which 13 identified as positive factors and 11 negative factors regarding their impact to scaling program. These factors are further categorized into the so-called SWOT matrix and based on this matrix successful strategies are identified for the organization. This model helps to examine the competitive position of the corporation by assessing the identified swots. This model will assist the GSD organizations to assess and measure their preparation preceding to the implementation of agile development.

2.4. What methods are mostly studied in distributed Agile?

From the resulted database, we identified a major contribution and discussion in regard of Scrum methodology. This method was studied in seven papers from a total of nine papers which had a focus on specific methodologies during the period of 2017-2019 and two papers from three during the period of 2020-2021. One paper was identified in regard of Lean Software Development and another paper was related to Pair Programming.

This major interest in Scrum relates to the fact that it is the most applied method in industry (Boyer and Mili, 2011). Initially considered for non-distributed projects, now Scrum needs to be adapted and customized based on global needs. Most proposals target changing components of the Scrum core procedures (Lous *et al*., 2017).

Razzak (2017) reported about the Lean Software development. They inform about how leanness encourages adaptability in distributed programming to accelerate improvement process.

Beecham *et al.*, (2021) in their a GSD case study examined two scaling agile frameworks; the Disciplined Agile Delivery (DAD) and the Scaled Agile Framework (SAFe). Both these frameworks put a great emphasis on risk mitigation, so it seemed suitable to the authors to develop a GSD Risk Catalog of 63 risks and then to evaluate their efficacy at tackling global software development project risks by studying how well they covered the software project risks identified the GSD Risk Catalog. It is concluded in the end that the two mentioned scaling agile frameworks address the 63 software development risks in the GSD Risk Catalog, so they can potentially eliminate or mitigate software project risks in global software development. Scrum of scrums is another scaled agile framework suggested to be taken into consideration for further studies in GSD.

3. CONCLUSIONS

We presented a literature review on Agile in Global Software Development with the focus on the real case studies provided by the industry. The study is based on a dataset extracted from the publications of the recent years 2017-2021. We found 46 publications in regard, which is a considerable amount based on the brief period. In these publications we identified reported problems and challenges, but also best practices that resulted in successful stories.

It cannot be generalized in all scenarios if Agile should be the best solution for all cases of distributed software development. Many aspects should be taken in consideration. Results show that distribution its-self poses difficulties, like the lack of face-to-face communication, language misunderstanding, but if the basic guidelines of Agile about the frequent meetings using communication tools and brainstorming are correctly followed, these difficulties can be overcome.

Also, other aspects need to be delved into. Hypotheses are raised on the impact of ethic, socio- cultural aspect, but these need further investigation. In addition, the impact that GSD has on specific roles of Agile are another open point that need further investigation. The present review, only the definition of Technical Lead could be identified.

Furthermore, we identified that scrum is mostly considered by the research, with new proposals of further investigation on lean methodology and scaled agile SAFe and DAD.

Our approach leaves room for extension, as the focus of this review was to identify the state of the art in the industry of the global software development using agile methodologies, and identify aspects need to be considered in future studies.

REFERENCES

Aggarwal A.K, Mani V.S, 2019. Using product line engineering in a globally distributed agile development team to shorten release cycles effectively. *ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE).* IEEE.

Awar K.B, Sameem M.SH.I, Hafeez Y, 2017. A model for applying Agile practices in Distributed environment: A case of local software industry. *International Conference on Communication, Computing and Digital Systems (C-CODE).* IEEE.

Bass J.M, Beecham S, Razzak M.A, Canna C.N, Noll J, 2018. An empirical study of the product owner role in scrum. In Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings (ICSE '18). Association for Computing Machinery, New York, NY, USA, 123–124.

Beecham S, Clear T, Lal R, Noll J, 2021. Do scaling agile frameworks address global software development risks? An empirical study. *Journal of Systems and Software*.

Bjørn P, Søderberg A-M, Krishna S, 2019. Translocality in global software development: The dark side of global agile. *Human–Computer Interaction 34.2*, (pp. 174-203).

Boyer J. A, Mili H, 2011. Agile business rule development. (pp. 49-71). Berlin: Springer.

Britto R, Smite D, Damm L-O, Börstler J, 2020. Evaluating and strategizing the onboarding of software developers in large-scale globally distributed projects. *Journal of Systems and Software*, Elsevier.

Calefato F, Ebert C, 2019. Agile Collaboration for Distributed Teams [Software Technology]. *IEEE Software 36.1* (pp. 72-78). IEEE.

Calefato F, Dubey A, Ebert C, Tell P, 2020. Global Software Engineering: Challenges and solutions. *Journal of Systems and Software.* Elsevier.

Camara R, Alves A, Monte I, Marinho M. L, 2020. Agile Global Software Development: A Systematic Literature Review. *SBES '20: Proceedings of the 34th Brazilian Symposium on Software Engineering*, (pp. 31-40).

Camara R, Alves A, Monte I, Marinho M, 2020. Agile Global Software Development: A Systematic Literature Review. *SBES '20: Proceedings of the 34th Brazilian Symposium on Software Engineering*, (pp. 31-40).

Carmel E, Espinosa J.A, Dubinsky Y, 2014. "Follow the Sun" Workflow in Global Software Development. *Journal of Management Information Systems 27.1*, (pp. 17-38).

Costa M.C.C, Lemos G.S, Beck F, 2018. Software engineering tools environment for outsourcing teams collaboration. *IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE).* IEEE.

Damian D.I, Izquierdo L., Singer J, Kwan I, 2007. Awareness in the wild: why communication breakdowns occur. *International Conference on Global Software Engineering*, (pp. 81–90). New Delhi, India.

Drechsler A, Breth S, 2019. How to go global: A transformative process model for the transition towards globally distributed software development projects. *International Journal of Project Management 37.8*, (pp. 941-955).

Ebert C, Kuhrmann M, Prikladnicki R, 2016. Global Software Engineering: Evolution and Trends. 11th International Conference on Global Software Engineering. Orange County, CA, USA, pp. 144-153, doi: 10.1109/ICGSE.2016.19.

Geeling S, Brown I, Weimann P, 2020. Cultural levels and emergent cultural contradictions in is development. *In Proceedings of the 28th European Conference on Information Systems (ECIS).* An Online AIS Conference.

Gupta R.K, Jain S, Singh B, Jha S.K, 2019, (b). Key Factors in Scaling up Agile Team in Matrix Organization. *Proceedings of the 12th Innovations on Software Engineering Conference (formerly known as India Software Engineering Conference).*

Gupta R.K, Venkatachalapathy M, Jeberla F. K, 2019, (a). Challenges in adopting continuous delivery and DevOps in a globally distributed product team: a case study of a healthcare organization. *ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE).* IEEE.

Humble J, 2018. Continuous delivery sounds great, but will it work here? *Communications of the ACM 61.4*, (pp. 34-39).

Inayat I, Marczak S, Salim S.S, Damian D, 2017. Patterns of collaboration driven by requirements in agile software development teams. *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer.

Kahya M.D, Seneler Ç, 2018, (a). Geographical Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company. *3rd International Conference on Computer Science and Engineering (UBMK).* IEEE.

Kahya M.D, Seneler C.O, 2018, (b) Temporal Distance Challenges in Distributed Agile Software Development: Case Study of a Global Company. 2nd International Symposium on Innovative Approaches in Scientific Studies

Kelly A, 2019. "Who Is the Product Owner?." The Art of Agile Product Ownership. (pp. 21-29). Apress, Berkeley, CA.

Khan A.A, Shameem M, Nadeem M, Akbar M.A, 2021. Agile trends in Chinese global software development industry: Fuzzy AHP based conceptual mapping. *Applied Soft Computing,Elsevier*.

Khmelevsky Y, Li X, Madnick S, 2017. Software development using agile and scrum in distributed teams. *Annual IEEE International Systems Conference (SysCon)*. pp. 1-4, IEEE.

Kroll J, Friboim S, Hemmati H, 2017. An empirical study of search-based task scheduling in global software development. *IEEE/ACM 39th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*, pp. 183-192.

Lous P, Kuhrmann M, Tell P. 2017. Is scrum fit for global software engineering?. *IEEE 12th International Conference on Global Software Engineering (ICGSE)*.

Lunesu M.I, Münch J, Marchesi M, Kuhrmann M, 2018. Using simulation for understanding and reproducing distributed software development processes in the cloud. *Information and Software Technology 103*, (pp. 226-238).

Majdenbaum A, Chaves M, 2020. Social interaction promotion in distributed software development in agile projects. *In Proceedings of the 28th European Conference on Information Systems (ECIS).* An Online AIS Conference.

Marinho M, Noll J, Richardson I, Beecham S, 2019. Plan-driven approaches are alive and kicking in agile global software development. *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM).* IEEE.

Martakis A, Deneva M, 2013. "Handling requirements dependencies in agile projects: a focus group with agile software development practitioners. *International Conference on Research Challenges in Information Science*, (pp. 1–11). Paris.

McCarthy S, O'Raghallaigh P, Fitzgerald C, Adam F, 2020. Building Bridges, Burning Bridges: The Use of Boundary Objects in Agile Distributed ISD Teams.

Moe N.B, Stray V, Goplen M.R, 2020. Studying Onboarding in Distributed Software Teams: A Case Study and Guidelines. *EASE '20: Proceedings of the Evaluation and Assessment in Software Engineering*, (pp. 150–159).

Paramartha M.A, 2018. *Requirements Engineering Issues In Agile Distributed Software Development*.

Penta M.D, Harman M, Antoniol G, 2011. The use of search-based optimization techniques to schedule and staff software projects: an approach and an empirical study. *In Software - Practice and Experience*, (pp. 495 – 519).

Putta A, 2018. Scaling agile software development to large and globally distributed large-scale organizations. *Proceedings of the 13th International Conference on Global Software Engineering.*

Qahtani A.M, 2020. Study of Agile Testing in A Distributed Software Development Project. *In Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis (ICGDA 2020).* Association for Computing Machinery, New York, NY, USA, 110–114.

Rajpal M, 2018. Effective distributed pair programming. *IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE).* IEEE.

Razzak M.A 2017. An Empirical Study on Leanness and Flexibility in Distributed Software Development. *arXiv preprint arXiv:1711.01097*.

Salameh A, Bass J, 2019. Spotify tailoring for B2B product development. *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). IEEE.* IEEE.

Santos E.W.d, Nunes I, 2017. Investigating the effectiveness of peer code review in distributed software development. *Proceedings of the 31st Brazilian Symposium on Software Engineering.*

Seckin I, Ovataman T, 2018. An empirical study on scrum application patterns in distributed teams. *Proceedings of the 13th International Conference on Global Software Engineering.*

Shafiq M, Zhang Q, Akbar M.A, Kamal T, Mehmood F, Riaz M.T, 2020. Towards successful global software development. *EASE '20: Proceedings of the Evaluation and Assessment in Software Engineering*, (pp. 445–450).

Shameem M, Kumar R.R, Nadeem R, Khan A.A, 2020. Taxonomical classification of barriers for scaling agile methods in global software development environment using fuzzy analytic hierarchy process. Applied Soft Computing. Elsevier.

Sinha R, Shameem M, Kumar C, 2020. SWOT: Strength, Weaknesses, and Threats for Scaling Agile Methods in Global Software Development. *Proceedings of the 13th Innovations in Software Engineering Conference on Formerly known as India Software Engineering Conference.*

Stray V, Moe N.B, 2020. Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *The Journal of Systems & Software, Elsevier*.

Szabó D.M, Steghöfer J-P, 2019. Coping strategies for temporal, geographical and sociocultural distances in agile GSD: A Case Study. *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP).* IEEE.

Uludağ Ö, Kleehaus M, Dreymann N, Kabelin C, Matthes F, 2019. Investigating the adoption and application of large-scale scrum at a German automobile manufacturer. *ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE).* IEEE, (pp. 22-29).

Vallon R, Estácio B.J.d.S, Prikladnicki R, Grechenig T, 2018. Systematic literature review on agile practices in global software development. *Information and Software Technology 96*, (pp. 161-180).

Vithana, V.N, Asirvatham D, Johar M.G.M, 2018. An Empirical Study on Using Agile Methods in Global Software Development. *18th International Conference on Advances in ICT for Emerging Regions (ICTer),* IEEE, (pp. 150-156).

Werewka J, Wietecha M, Wolczyk K, 2017. Distinguishing and defining the role of a technical leader in outsourced teams developing IT solutions using Scrum. *FedCSIS Communication Papers.*