The Impact of Human Factors in Distributed Software Engineering

Austin D.T. FitzGerald

University of Wisconsin – Platteville

INDSTENG 3430: Human Factors Engineering

Dr. Dianne McMullin

November 25, 2020

1) Introduction

The advent of the COVID-19 pandemic has forced software engineers across the world to shift into remote work and their dichotomous experiences are already making their way into publications (Ford et al., 2020; Ralph et al., 2020). The scale of this shift is unlike what has ever been seen, but literature involving the analysis and review of distributed software development is not as new. The evolution of technology has dictated globalization of markets, especially in software development (Prikladnicki et al., 2003). As a result, research regarding software engineering in a distributed context has increased (Marques et al., 2012). The purpose of this literature review is to synthesize concepts provided in recent research and provide an overview of the impact of human factors on software quality in distributed environments.

The process of gathering literature was carried out using the following parameters.

1. Studies published since 2000 were considered for review. This choice of year was arbitrary but is intended to limit the scope of this review to the most recently accepted work.
2. Studies published in the English language were considered for review. The quality of grammar was used as a secondary factor for consideration but was not intended to limit scope to research conducted in Western culture.
3. Systematic reviews and mapping studies were leveraged but were not used as direct sources unless they made unique contributions.

Within these parameters, the following databases and catalogs were utilized: ACM Digital Library, arXiv, Gale, Google Scholar, IEEE Xplore, ResearchGate, ScienceDirect, Search@UW, SpringerLink, and Wiley Online Library. Searches included various combinations of the following keywords: (dislocated, dispersed, distributed, global, nearshore, offshore, offsite, virtual, work from home) and (quality, security, software) and (development, engineering, organizations, teams) and (effects, failure, impacts, improvement) and (aspects, awareness, barriers, capabilities, communication, coordination, culture, factors, happiness, human, impediments, information sharing, obstacles, sociotechnical) and (case study, empirical, review, taxonomy). Zotero was used as a reference-management tool.

2) Terminology of Distributed Software Engineering

Distributed software engineering (DSE), also referred to as distributed software development (DSD), is a growing field of study that has no unified terminology or taxonomy used by researchers (Šmite et al., 2014). Many terms may overlap meaning across different papers, but this study (Šmite et al., 2014) found that terminology was rarely defined by authors. This leaves even the most basic definitions unstandardized, such as that of distributed software engineering. Both (Gumm, 2006) and (Šmite et al., 2014) reviewed existing literature to describe DSE as a facet of geographic location and relationships involving inter-organizational and intra-organizational structures, with (Šmite et al., 2014) further becoming the first to map an empirically based taxonomy. While general, this definition does not delineate the intentions of all authors. For example, (Šmite et al., 2014) describes global software engineering/development (GSE/GSD) as equivalent to DSE. In contrast, it is common for authors to define GSE as a distinct subset of DSE, where software engineering is distributed across multiple countries (Bird et al., 2009; Gumm, 2006; Herbsleb, 2007; Jalali & Wohlin, 2010; Marques et al., 2012; Prikladnicki et al., 2003, 2007). Furthermore, some papers state that culture may not only be a factor related to geographic location, it may also be a distinct dimension of DSE (Al-Ani & Edwards, 2008; Conchuir et al., 2006; Rajpal, 2018). Both (Gumm, 2006) and (Šmite et al.,

2014) recognize the impact of culture on DSE, but do not offer any explanation for the lack of its inclusion as a dimension separate from geographic location. Likewise, a case study addressing the human factors considered important by software engineers indicated that current theories in software engineering need to consider more dimensions (P. Lenberg et al., 2015).

The taxonomy provided in (Šmite et al., 2014) divides DSE into four levels: location (onshore or offshore), legal entity (insourcing or outsourcing), geographic distance (close or distant), and temporal distance (similar or different). This taxonomy is useful in classifying existing studies involving DSE which use non-standard terms to describe their organizational structure. Common terms such as dispersed, distributed, remote, and virtual software engineering teams are examples of low-level organizational structures that are not defined in existing taxonomies for DSE. In (Bird et al., 2009), a distributed team is defined as a group of individuals which are geographically distributed among different campuses who work on the same executable for a project. Contrarily, this description of a distributed team in (Bird et al., 2009) is more akin to that of a virtual team in (Jalali & Wohlin, 2010), whereby virtual team members work "jointly on the same tasks" and distributed team members do not. Yet, in other papers, the terms are used interchangeably as a general description of teams which are project-specific and contain any members who are geographically distributed (Andres, 2002; Gumm, 2006; Olariu & Aldea, 2014; Olson & Olson, 2000; Pierce & St.Amant, 2011; Reed & Knight, 2010). Nevertheless, the taxonomy in (Šmite et al., 2014) can be used to classify all DSE studies in this literature review, even those which use non-standard terms. That is, the human factors researched in a global context also apply to that of DSE with shorter geographic distance.

3) Human Factors Research in Traditional Software Engineering

Software engineering (SE) is a sociotechnical activity entrenched in human-centered design. In this context, human-centered design is a sustainable approach in SE which addresses ergonomics regarding users, developers, and other stakeholders (International Organization for Standardization, 2010). As a human-centered activity, each phase of SE is highly affected by the feelings, attitudes, and behaviors of customers and stakeholders alike (Acuna et al., 2006; Per Lenberg et al., 2014). Yet, researchers have only just begun to unravel the impact of human factors in SE despite their critical role (Capretz, 2014). Longstanding overrepresentation of technical aspects has obstructed the advancement of software engineering practices in the classroom, lab, and workplace (Capretz, 2014; Feldt et al., 2008; Pirzadeh, 2010). This is most evident when considering that *The Guide to the Software Engineering Body of Knowledge (SWEBOK)*, a book which is listed as an international standard and is a foundation for updating accreditation criteria for universities, only briefly mentions human factors in SE (Bourque et al., 2014; Fairley et al., 2014; International Organization for Standardization, 2015). In what (Amrit et al., 2014) considers to be the only comprehensive systematic review of human factors in SE, (Pirzadeh, 2010) concluded that human factors have been overlooked by primary study researchers. This is not an isolated observation. Recently, the number of papers related to human factors in software engineering is on an upward trend despite the failure of researchers in providing accurate and searchable titles/keywords/abstracts (Amrit et al., 2014).

4) Human Factors Research in Distributed Software Engineering

A distributed environment has direct impacts on the ways which software is designed, implemented, tested, and delivered. Theses impacts come by way of differences in communication, coordination, and cooperation when compared to the traditional software engineering process (Šmite et al., 2008). These three terms are used widely throughout papers involving distributed software engineering and are likened as the “Three C’s” or the “3C Model”. The following definitions are adapted from two papers. The first, (Fuks et al., 2007), attempts to generalize a collaborative systems. The second, (Mishra & Mishra, 2009), is a paper which explicitly provides definitions in the context of software engineering; it uses the term collaboration instead of cooperation, as do some other papers. Similarly, to general DSE terminology, most researchers do not provide definitions or references for this terminology. Communication is the exchange of information via any media between people. Coordination is the organization of people and tasks. Cooperation is the process of people working together on tasks in a shared environment. Human factors involved at personal, interpersonal, and organizational levels of DSE have interrelated impacts on each of the Three C’s. Culture is a factor which is discussed in most literature referenced in this review. It can impact each of the Three C’s due to personal, interpersonal or organizational reasons (Prikladnicki et al., 2003; Schloegel et al., 2018). Overall, there seems to be a lack of research specifically involving the personal factors impacting DSE, many of the changes compared to traditional SE are largely interpersonal and organizational. Recent surveys sent to software engineers regarding their opinions on shifts to remote work because of the COVID-19 pandemic may be applicable. Two of the personal factors associated with large reductions in productivity in (Ford et al., 2020) are lack of motivation and poor home work environment. Both (Ford et al., 2020) and (Ralph et al., 2020) recognize that factors related to distributed work during a pandemic are not shown to be applicable to DSE as a whole. Lastly, interpersonal factors that effect all Three C’s are trust and awareness (Espinosa et al., 2007; Omoronyia et al., 2010; Trainer & Redmiles, 2018).

4.A) Trust and Awareness

Both trust and awareness have been greatly explored in the context of DSE. In an article titled *Bridging the Gap Between Awareness and Trust in Globally Distributed Software Teams*, (Trainer & Redmiles, 2018) define interpersonal trust in DSE as the positive or negative expectations that people have about each other's behavior. A person's perceived trustworthiness is built on some attributions, ways that people apply reason to the cause of events (Trainer & Redmiles, 2018). Furthermore, a fundamental attribution error occurs when a person makes an attribution based on someone's personal characteristics when they should have instead based it on the outside circumstances surrounding the event. As noted in (Casey, 2010), fundamental attribution errors are almost natural in DSE due to the lack of social presence which allows for more situational context, trust must be "actively facilitated, fostered and developed".

Awareness is a related concept that is multi-dimensionally defined in (Omoronyia et al., 2010) as follows. Informal awareness involves knowing what group members are doing. Group-structural awareness involves knowing the roles of group members. Social awareness involves knowing the state of attention, interests, and emotions of group members. Workspace awareness involves knowing the resources related to coordination. Furthermore, (Omoronyia et al., 2010) and (Trainer & Redmiles, 2018) separately describe how awareness is contextualized over time such that deeper levels of a person's understanding of their work environment and teammates is formed. This contextual awareness is helpful in exploring another group member's situation, and therefore lead to more accurate attributions (Trainer & Redmiles, 2018). Developing and maintaining awareness is critical for establishing trust, but difficult in DSE (Dullemond et al., 2012).

4.B) Communication, Coordination, and Collaboration

According to (Bird et al., 2009), communication is the single most referenced factor related to challenges in DSE. Traditional face-to-face meetings are, by definition of DSE, less common. So, most communication is required to use some technological media. (Andres, 2002) describes the need for adequate communications mechanisms in DSE in terms of media richness in relation to socio-emotional communication. That is, communication is best impacted by rich media, such as videoconferencing, due to the social presence which is required to reduce ambiguity. (Herbsleb et al., 2000) found that the loss of face-to-face communication introduced significant delay in cross-site work due to inadequate tools available for rich communication. However, (Olson & Olson, 2000) states that attributes of DSE, specifically distance and cultural differences, cause profoundly negative impacts on the effectiveness of communication no matter the technology that is used. Similarly, (Reed & Knight, 2010) concluded that inadequate communication was found to be a risk, but not due to technological problems hindering it. Other papers argue that the human factors of communication are more impactful than the tools used to do so, but that tools do have some impact (Henshel et al., 2015; Jiménez et al., 2009; Jolak et al., 2018; Stray & Moe, 2020). In contrast, some research has suggested that geographic distribution does not have a significant impact on the effectiveness of communication (Bird et al., 2009; T. Nguyen et al., 2008). There are some contradictory results from studies which explain organizational factors that impact communication in DSE, but each paper generally extends observations that interpersonal factors are decisive. Greater awareness and trust in team members has been shown to mitigate geographic, temporal, and cultural differences that negatively impact communication (Bird et al., 2009; Omoronyia et al., 2010). Likewise, having negative trust in managers has been observed to decrease the effectiveness of communication (Casey, 2010). In the cybersecurity realm, (Henshel et al., 2015) describes the importance of leveraging communication in a timely manner and using secure communication tools. Greater awareness of defined communication models has been suggested to positively impact communication as well (Al-Ani & Edwards, 2008).

As stated in (Herbsleb, 2007), the key phenomenon of DSE is coordination over distance. There can be any number of individuals or teams that are owners of certain parts of a software project. For software parts to eventually compose an entire product, individuals and teams must share schedules, allocate responsibilities, and maintain dependencies of tasks onto individuals and teams (Espinosa et al., 2007). Therefore, coordination leverages communication and cooperation by engaging people in meetings purposed to create schedules, responsibilities, and tasks (Fuks et al., 2007; Jiménez et al., 2009; Misra & Fernández-Sanz, 2011). In practice, coordination is largely dependent on the engineering approaches used. In the general case of DSE, multiple studies have concluded that it is more difficult to coordinate tasks across geographically and temporally dispersed sites (Herbsleb et al., 2000; Olson & Olson, 2000; Prikladnicki et al., 2003; Stray & Moe, 2020; Suali et al., 2017). Most papers reference issues in communication as major reasons for coordination breakdowns. Yet, like for communication, some papers argue against any significant relationship between geographic distance and effective coordination (Bird et al., 2009; T. Nguyen et al., 2008; Stray & Moe, 2020). Some researchers have observed that organizational support for coordination structures needs to be focused on early in the project timeline. Software architects, who begin to work in the earliest stages of development, have a major role in shaping the mechanisms for coordination (Herbsleb, 2007). It is vital to leverage early cooperation of developers and stakeholders in defining involved parties, expectations for coordination, scheduling, and other mechanisms (Casey, 2010; Herbsleb, 2007; Stray & Moe, 2020). Other organizational factors impacting coordination are most relevant when considering teams which operate using SE methodologies such as Agile, which are designed with specific human roles, timelines, and meetings (Cockburn & Highsmith, 2001). One study specifically references the struggles that come with coordinating security objectives in an Agile DSE setting (van der Heijden et al., 2018). As an interpersonal factor, the development of contextual awareness is vital for effective coordination (Espinosa et al., 2007; Lous et al., 2018). Thus some studies describe processes called information sharing which are designed to purposefully engage team members in formal and informal communication (Espinosa et al., 2007; Herbsleb, 2007; Jolak et al., 2018; Misra et al., 2013; Stray & Moe, 2020).

The last of the 3 C's is cooperation, called collaboration by some, describes actions that most people probably think of when they consider SE. Processes in cooperation should be directly be supported by the coordination of responsibilities and tasks (Jiménez et al., 2009). Cooperation requires communication (Fuks et al., 2007). Cooperation is essential for the effective operation of virtual teams, but geographical and temporal distance has been shown to negatively impact the ability to cooperate (Casey, 2010; Misra et al., 2013). Yet, (Lous et al., 2018) makes a contrary claim; traditional co-located work environments are not the ideal setup for group work in software engineering. That is, the continuous improvement of processes and tools is dictated by the ways which groups want to cooperate. There are plenty of existing tools to leverage in distributed cooperation, and some studies have claimed more importance of consistency in choosing tools than the specific tools themselves (Bird et al., 2009; Prikladnicki et al., 2003; Rajpal, 2018; Stray & Moe, 2020). Likewise, multiple studies have shown success in DSE after careful and collaborative implementation of SE methodologies such as Agile and eXtreme Programming, as well as techniques like pair programming (Cockburn & Highsmith, 2001; Mishra & Mishra, 2009; Rajpal, 2018).

5) Impacts of DSE on Software Quality

Group sizes in DSE have been observed to be larger than the traditional SE counterparts (Al-Ani & Edwards, 2008; Stray & Moe, 2020). Multiple studies have found that defects are more commonly found in code worked on by a large number of developers (Bird et al., 2009; Meneely et al., 2013; T. H. D. Nguyen et al., 2016). However, both (Bird et al., 2009) and (T. H. D. Nguyen et al., 2016) find that DSE has no significant effect on the quality of software artifacts after controlling for team size. As stated in the previous sections, human factors which negatively impact communication, coordination, or cooperation could also impact the quality of the software produced. Unfortunately, as far as this literature review is concerned, there are very few studies which leverages both human factors in DSE and software quality metrics. Therefore, most papers are exploratory in nature and do not contain conclusions which go beyond what is stated in the previous sections (Jabangwe et al., 2015; Misra & Fernández-Sanz, 2011). Software quality can be positively and negatively impacted by geographic distance, temporal distance, cultural differences, organizational structure relationships, communication, coordination, cooperation, awareness, trust, knowledge sharing, and development methodologies.

Acuna, S. T., Juristo, N., & Moreno, A. M. (2006). Emphasizing human capabilities in software development. *IEEE Software*, *23*(2), 94–101. https://doi.org/10.1109/MS.2006.47

Al-Ani, B., & Edwards, H. K. (2008). A Comparative Empirical Study of Communication in Distributed and Collocated Development Teams. *2008 IEEE International Conference on Global Software Engineering*, 35–44. https://doi.org/10.1109/ICGSE.2008.9

Amrit, C., Daneva, M., & Damian, D. (2014). Human factors in software development: On its underlying theories and the value of learning from related disciplines. A guest editorial introduction to the special issue. *Information and Software Technology*, *56*(12), 1537–1542. https://doi.org/10.1016/j.infsof.2014.07.006

Andres, H. (2002). A comparison of face-to-face and virtual software development teams. *Team Performance Management*, *8*, 39–48. https://doi.org/10.1108/13527590210425077

Bird, C., Nagappan, N., Devanbu, P., Gall, H., & Murphy, B. (2009). Does distributed development affect software quality? An empirical case study of Windows Vista. *2009 IEEE 31st International Conference on Software Engineering*, 518–528. https://doi.org/10.1109/ICSE.2009.5070550

Bourque, P., Fairley, R. E., & IEEE Computer Society. (2014). *Guide to the software engineering body of knowledge*.

Capretz, L. F. (2014). Bringing the Human Factor to Software Engineering. *IEEE Software*, *31*(2), 104–104. https://doi.org/10.1109/MS.2014.30

Casey, V. (2010). Developing Trust In Virtual Software Development Teams. *Journal of Theoretical and Applied Electronic Commerce Research*, *5*(2), 41–58. https://doi.org/10.4067/S0718-18762010000200004

Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, *34*(11), 131–133. https://doi.org/10.1109/2.963450

Conchuir, E. O., Holmstrom, H., Agerfalk, P. J., & Fitzgerald, B. (2006). Exploring the Assumed Benefits of Global Software Development. *2006 IEEE International Conference on Global Software Engineering (ICGSE'06)*, 159–168. https://doi.org/10.1109/ICGSE.2006.261229

Dullemond, K., Gameren, B. van, & Solingen, R. van. (2012). Supporting distributed software engineering in a fully distributed organization. *2012 5th International Workshop on Co-Operative and Human Aspects of Software Engineering (CHASE)*, 30–36. https://doi.org/10.1109/CHASE.2012.6223017

Espinosa, J., Slaughter, S., Kraut, R., & Herbsleb, J. (2007). Team Knowledge and Coordination in Geographically Distributed Software Development. *J. of Management Information Systems*, *24*, 135–169. https://doi.org/10.2753/MIS0742-1222240104

Fairley, R. E. D., Bourque, P., & Keppler, J. (2014). The impact of SWEBOK Version 3 on software engineering education and training. *2014 IEEE 27th Conference on Software*

Engineering Education and Training (CSEE T), 192–200. https://doi.org/10.1109/CSEET.2014.6816804

Feldt, R., Torkar, R., Angelis, L., & Samuelsson, M. (2008). Towards individualized software engineering: Empirical studies should collect psychometrics. *Proceedings of the 2008 International Workshop on Cooperative and Human Aspects of Software Engineering - CHASE '08*, 49–52. https://doi.org/10.1145/1370114.1370127

Ford, D., Storey, M.-A., Zimmermann, T., Bird, C., Jaffe, S., Maddila, C., Butler, J. L., Houck, B., & Nagappan, N. (2020). A Tale of Two Cities: Software Developers Working from Home During the COVID-19 Pandemic. *ArXiv:2008.11147 [Cs]*. http://arxiv.org/abs/2008.11147

Fuks, H., Raposo, A., Gerosa, M. A., Pimental, M., & Lucena, C. (2007). *The 3C collaboration model* (pp. 637–644). https://doi.org/10.4018/978-1-59904-000-4.ch097

Gumm, D. C. (2006). Distribution Dimensions in Software Development Projects: A Taxonomy. *IEEE Software*, *23*(5), 45–51. https://doi.org/10.1109/MS.2006.122

Henshel, D., Cains, M. G., Hoffman, B., & Kelley, T. (2015). Trust as a Human Factor in Holistic Cyber Security Risk Assessment. *Procedia Manufacturing*, *3*, 1117–1124. https://doi.org/10.1016/j.promfg.2015.07.186

Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. *Future of Software Engineering (FOSE '07)*, 188–198. https://doi.org/10.1109/FOSE.2007.11

Herbsleb, J. D., Mockus, A., Finholt, T. A., & Grinter, R. E. (2000). Distance, dependencies, and delay in a global collaboration. *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work - CSCW '00*, 319–328. https://doi.org/10.1145/358916.359003

International Organization for Standardization. (2010). *Ergonomics of human-system interaction—Part 210: Human-centred design for interactive systems* (ISO Standard No. 9241-210:2010). https://www.iso.org/standard/52075.html

International Organization for Standardization. (2015). *Software Engineering—Guide to the software engineering body of knowledge (SWEBOK)* (ISO/IEC Standard No. 19759:2015). https://www.iso.org/standard/67604.html

Jabangwe, R., Börstler, J., & Petersen, K. (2015). Handover of managerial responsibilities in global software development: A case study of source code evolution and quality. *Software Quality Journal*, *23*(4), 539–566. https://doi.org/10.1007/s11219-014-9247-1

Jalali, S., & Wohlin, C. (2010). Agile Practices in Global Software Engineering—A Systematic Map. *2010 5th IEEE International Conference on Global Software Engineering*, 45–54. https://doi.org/10.1109/ICGSE.2010.14

Jiménez, M., Piattini, M., & Vizcaíno, A. (2009). Challenges and Improvements in Distributed Software Development: A Systematic Review. *Advances in Software Engineering*, *2009*, 1–14. https://doi.org/10.1155/2009/710971

Jolak, R., Wortmann, A., Chaudron, M., & Rumpe, B. (2018). Does Distance Still Matter? Revisiting Collaborative Distributed Software Design. *IEEE Software*, *35*(6), 40–47. https://doi.org/10.1109/MS.2018.290100920

Lenberg, P., Feldt, R., & Wallgren, L. G. (2015). Human Factors Related Challenges in Software Engineering – An Industrial Perspective. *2015 IEEE/ACM 8th International Workshop on Cooperative and Human Aspects of Software Engineering*, 43–49. https://doi.org/10.1109/CHASE.2015.13

Lenberg, Per, Feldt, R., & Wallgren, L.-G. (2014). Towards a behavioral software engineering. *Proceedings of the 7th International Workshop on Cooperative and Human Aspects of Software Engineering*, 48–55. https://doi.org/10.1145/2593702.2593711

Lous, P., Tell, P., Michelsen, C. B., Dittrich, Y., Kuhrmann, M., & Ebdrup, A. (2018). Virtual by Design: How a Work Environment can Support Agile Distributed Software Development. *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*, 97–106.

Marques, A. B., Rodrigues, R., & Conte, T. (2012). Systematic Literature Reviews in Distributed Software Development: A Tertiary Study. *2012 IEEE Seventh International Conference on Global Software Engineering*, 134–143. https://doi.org/10.1109/ICGSE.2012.29

Meneely, A., Srinivasan, H., Musa, A., Tejeda, A. R., Mokary, M., & Spates, B. (2013). When a Patch Goes Bad: Exploring the Properties of Vulnerability-Contributing Commits. *2013 ACM / IEEE International Symposium on Empirical Software Engineering and Measurement*, 65–74. https://doi.org/10.1109/ESEM.2013.19

Mishra, D., & Mishra, A. (2009). Effective communication, collaboration, and coordination in eXtreme Programming: Human-centric perspective in a small organization. *Human Factors and Ergonomics in Manufacturing*, *19*(5), 438–456. https://doi.org/10.1002/hfm.20164

Misra, S., Colomo-Palacios, R., Pusatlı, T., & Soto-Acosta, P. (2013). A discussion on the role of people in global software development. *Tehnicki Vjesnik*, *20*, 525–531.

Misra, S., & Fernández-Sanz, L. (2011). Quality Issues in Global Software Development. *ICSEA 2011: The Sixth International Conference on Software Engineering Advances*, 325–330.

Nguyen, T. H. D., Adams, B., & Hassan, A. E. (2016). Does Geographical Distance Effect Distributed Development Teams: How Aggregation Bias in Software Artifacts Causes Contradictory Findings. *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)*, 412–423. https://doi.org/10.1109/ISSRE.2016.36

Nguyen, T., Wolf, T., & Damian, D. (2008). Global Software Development and Delay: Does Distance Still Matter? *2008 IEEE International Conference on Global Software Engineering*, 45–54. https://doi.org/10.1109/ICGSE.2008.39

Olariu, C., & Aldea, C. C. (2014). Managing Processes for Virtual Teams – A BPM Approach. *Procedia - Social and Behavioral Sciences*, *109*, 380–384. https://doi.org/10.1016/j.sbspro.2013.12.476

Olson, G. M., & Olson, J. S. (2000). Distance Matters. *Human–Computer Interaction*, *15*(2–3), 139–178. https://doi.org/10.1207/S15327051HCI1523_4

Omoronyia, I., Ferguson, J., Roper, M., & Wood, M. (2010). A review of awareness in distributed collaborative software engineering. *Software: Practice and Experience*, *40*(12), 1107–1133. https://doi.org/10.1002/spe.1005

Pierce, R., & St.Amant, K. (2011). Working from home in a globally distributed environment. *SIGDOC'11 - Proceedings of the 29th ACM International Conference on Design of Communication*. https://doi.org/10.1145/2038476.2038519

Pirzadeh, L. (2010). *Human Factors in Software Development: A Systematic Literature Review* [Master's thesis, Chalmers University of Technology]. https://hdl.handle.net/20.500.12380/126748

Prikladnicki, R., Audy, J. L. N., Damian, D., & Oliveira, T. C. de. (2007). Distributed Software Development: Practices and challenges in different business strategies of offshoring and onshoring. *International Conference on Global Software Engineering (ICGSE 2007)*, 262–274. https://doi.org/10.1109/ICGSE.2007.19

Prikladnicki, R., Audy, J. L. N., & Evaristo, R. (2003). Global software development in practice lessons learned. *Software Process: Improvement and Practice*, *8*(4), 267–281. https://doi.org/10.1002/spip.188

Rajpal, M. (2018). Effective Distributed Pair Programming. *2018 IEEE/ACM 13th International Conference on Global Software Engineering (ICGSE)*, 6–10. https://dl.acm.org/doi/10.1145/3196369.3196388

Ralph, P., Baltes, S., Adisaputri, G., Torkar, R., Kovalenko, V., Kalinowski, M., Novielli, N., Yoo, S., Devroey, X., Tan, X., Zhou, M., Turhan, B., Hoda, R., Hata, H., Robles, G., Fard, A. M., & Alkadhi, R. (2020). Pandemic Programming: How COVID-19 affects software developers and how their organizations can help. *Empirical Software Engineering*, *25*(6), 4927–4961. https://doi.org/10.1007/s10664-020-09875-y

Reed, A. H., & Knight, L. V. (2010). Effect of a virtual project team environment on communication-related project risk. *International Journal of Project Management*, *28*(5), 422–427. https://doi.org/10.1016/j.ijproman.2009.08.002

Schloegel, U., Stegmann, S., van Dick, R., & Maedche, A. (2018). Age stereotypes in distributed software development: The impact of culture on age-related performance expectations. *Information and Software Technology*, *97*, 146–162. https://doi.org/10.1016/j.infsof.2018.01.009

Šmite, D., Moe, N. B., & Torkar, R. (2008). Pitfalls in Remote Team Coordination: Lessons Learned from a Case Study. In A. Jedlitschka & O. Salo (Eds.), *Product-Focused Software Process Improvement* (pp. 345–359). Springer. https://doi.org/10.1007/978-3-540-69566-0_28

Šmite, D., Wohlin, C., Galviņa, Z., & Prikladnicki, R. (2014). An empirically based terminology and taxonomy for global software engineering. *Empirical Software Engineering*, *19*(1), 105–153. https://doi.org/10.1007/s10664-012-9217-9

Stray, V., & Moe, N. B. (2020). Understanding coordination in global software engineering: A mixed-methods study on the use of meetings and Slack. *Journal of Systems and Software*, *170*, 110717. https://doi.org/10.1016/j.jss.2020.110717

Suali, A. J., Fauzi, S. S. M., Sobri, W. A. W. M., & Nasir, M. H. N. M. (2017). Developers' coordination issues and its impact on software quality: A systematic review. *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 659–663. https://doi.org/10.1109/ICSITech.2017.8257195

Trainer, E. H., & Redmiles, D. F. (2018). Bridging the gap between awareness and trust in globally distributed software teams. *Journal of Systems and Software*, *144*, 328–341. https://doi.org/10.1016/j.jss.2018.06.028

van der Heijden, A., Broasca, C., & Serebrenik, A. (2018). An empirical perspective on security challenges in large-scale agile software development. *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*, 1–4. https://doi.org/10.1145/3239235.3267426