Social Network Multiplexity in Software Development: Investigating the Implications for Knowledge Sharing, Problem-Solving and Performance

A Joint Thesis Submitted in Fulfilment of the Requirements
For the Degree of Doctor of Philosophy

to the J.E. Cairnes School of Business and Economics
at the National University of Ireland, Galway

and

Institute for Management Research
Business Administration Strategy
Radboud University, The Netherlands

Matthew Oladeji Ajimati BBA, BBS, MBA, MSc

Primary Supervisors
Dr Eoin Whelan (NUIG)
Dr Rick Aalbers (RU)

Secondary Supervisors
Dr Lorraine Morgan (NUIG)
Prof (Dr) Hans van Kranenburg (RU)

January 2022

Table of Contents

List of Figures

List of Tables

Declaration

I declare that this thesis has not been submitted in any form for another degree at the National University of Ireland Galway and Radboud University, the Netherlands, or any other university. All research contained herewith is entirely my own and the use of all material from other sources has been properly and fully acknowledged.

………… . ………….

Signature Date

Abstract

Social structures matter to software development. This is because software developers need to collaborate to share knowledge and solve problems together to drive software development success. Nonetheless, studies investigating how such interaction affects performance is limited to a dominant perspective of technical advice network (one-dimensional relations) and less of business advice network, and how they interact in online and offline communication combined to determine the performance of software developers. A few scholars in information systems (IS) and software engineering (SE) fields have called for the application of a multiplex network perspective (multidimensional relations) as a theoretical lens to investigate the structure of software developers. This thesis applies a multiplex network perspective to explain individuals' embeddedness in social networks and what it means for performance-related outcomes based on three (interrelated) research studies.

The study presented in chapter 2 (article 1 related) focuses on understanding, conceptualising, and applying the concept of multiplexity to studying social networks in software development settings. A systematic review of social network analysis (SNA) within the IS, SE and management science domain was conducted. The findings revealed an isolated perspective of multiplexity based on a resource access stance (business/technical knowledge), communication (offline/online networks) and relationship (formal/informal relations) dimensions. The lack of an integrated perspective limits the ability to generate sufficient insights concerning the dynamics of the social network of developers, particularly how different interpersonal network configurations (ties) and their interrelations impact performance differently. This is often needed to make sense of inherent variations in network behaviours undergirding an individual's ability to generate and keep a steady flow of diverse resources.

The study presented in chapter 3 (article 2 related) focuses on the multiplex network structures of software developers by examining how their brokerage roles (gatekeeping, consulting etc.) and knowledge sharing interactions (business/technical

related) in the communication networks (offline/online) contribute to their software development effort. Prior studies have provided limited understanding about how the role and type of knowledge flowing through offline and online communication combined shape performance. Because studying diverse network structures and network knowledge types can generate understanding about an individual's leverage in problem-solving activities. This study thereby revealed the influence of different network behaviours and knowledge interactions in shaping performance potentials.

The study presented in chapter 4 (article 3 related) focuses on understanding the embedding of software developers in multiplex advice structures and how their brokerage in both business and technical advice networks provides access to business and technical knowledge resources and translate to an individual's problem-solving competence. Research evidence in this area offers a limited understanding of the network antecedents of problem-solving competence. This study revealed the varying level of significance of brokerage and contact quality within and between the structure of advice networks for knowledge resources and the problem-solving competence of software developers.

Overall, this thesis makes a number of contributions by integrating different dimensions of multiplexity, which have been observed in isolation in prior streams of literature and different fields, to generate sufficient insights about software development networks. In terms of contributions to theory, this research is one of the first to provide an integrated framework and its application to an individual's coordination of resources and task performance in software development. It also makes this study one of the first to clearly reveal the optimal and sub-optimal use of offline and online networks based on different knowledge interactions. This research is also one of the first to directly link business and technical advice connectedness via brokerage (as antecedents) to problem-solving competence (as outcome) of software developers. As per contributions to practice, the insights generated can help organisational managers and leaders to promote knowledge networking, cultivate strategic brokers, promote expert interventions, and optimise knowledge resourcefulness for greater problem-solving competence of software developers.

Building upon network theory or social network theory such as structural hole theory, the application of multiplexity to software development generate insights into the knowledge sharing processes and problem-solving potentials of software developers. In doing so, it provides a complete picture of the variance in brokerage action, network structure, and network knowledge influencing software development performance. In other words, it reveals the combined effect of the role of individuals, structures and knowledge types plays in deriving a better software development effort. This thesis considers these findings and their implications for future studies.

Keywords: software development, multiplexity, social network, problem-solving, brokerage, software developer, communication network, business knowledge, technical knowledge, performance

Dedications

I owe it to God for His loving kindness, faithfulness, and provisions throughout the journey of this PhD. The road was bumpy, but He saw me through it.

I dedicate this work to God Almighty and my Lord and Saviour, Jesus Christ, who never fails to strengthen me in my challenges and has made my joy to be full.

Acknowledgements

"*If you want to go fast, go alone. If you want to go far, go together*."

African Proverb

The ability to achieve successful completion of this thesis is no doubt, a result of the network of people whose presence, support, and guidance personally and professionally have contributed to this success.

I sincerely and especially appreciate the supervisory guidance of my primary supervisors: Dr Eoin Whelan and Dr Rick Aalbers, for sharing their time, effort, and expert knowledge with me. The ability to draw from your experience and knowledge in information systems and management science, respectively, has added value to my life. I cannot but thank Dr Eoin Whelan for seeing potential in me while I was an MSc student of his at the National University of Ireland Maynooth (NUIM) and has encouraged and allowed me to build a PhD career with him. The invaluable knowledge, insights and inputs of Dr Lorraine Morgan and Prof (Dr) Hans van Kranenburg have been very impactful. For Dr Lorraine, I am glad I got to know you also right from my time at NUIM, in helping to guide the then research proposal development. I'm so grateful! In addition, I must say to Dr Rick and Prof (Dr) Hans that your warm reception during my few visits to Radboud University is second to none and am grateful for it also.

My special thanks also to my graduate research committee: Dr Murray Scott, Dr Michael Lang, and Dr Anatoli Nachev, for their knowledge guidance through constructive reviews and feedback of my work to ensure progression in the right direction. Your recommendations at different times have been significant.

I am grateful to Dr Noel Carroll (Internal Examiner), Dr Michael Cahalane (External Examiner) from the University of New South Wales and Professor Kieran Conboy (Chair) for the Viva defence. Their questions and recommendations have been spotted, incorporated, and provided considerations for future research ambitions. This is now a much-improved thesis. Thank you!

Thanks to those who have also shared their practical knowledge and wisdom on navigating this journey the likes of Prof Kieran Conboy, Prof William Golden, Dr Martin Hughes, and Dr Denis Dennehy. My special thanks to you Dr Denis for going beyond professional support in sharing some beautiful baby gifts from your daughter and wife since the birth of our children (Dunamis and Treasure), some of which keep them busy for daddy when cracking with research work ☺.

Thanks again to Dr Lorraine for organising IS Seminar series, this has also birthed creativity in a few things I have done in this work. It has provided me with a good platform to learn new things from our numerous guests and researchers within the discipline. Please keep the initiative alive, very good for the motivation of PhD students, particularly getting to know ourselves sharing our highs and lows moments and drawing strengths together. Good times were learning and socialising with Azka Umair, Dr Mairead O Connor, Ronan Doyle etc.

The efforts of those who have assisted in the data collection process, in the person of Dr Eoin Higgins and Mr Barry Reddan, cannot be over-emphasised. I appreciate your genuine and generous sacrifice despite the disruptive nature of the pandemic.

I thank the support of Science Foundation Ireland grant 13/RC/2094_P2 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational Programme to Lero - the Science Foundation Ireland Research Centre for Software (www.lero.ie), National University of Ireland Galway, and co-funded by Radboud University, the Netherlands. I also thank the Higher Education Authority for the costed extension grant received to fill the gap meted by COVID-19. Your sources of funding for the PhD scholarship have fuelled this achievement.

My thanks also go to my family friends for their spiritual, moral, and generous support during this time. In persons of Pastor Esther Bolarin, Pastor Funmi Ogunsanwo, Mrs Felicia Akinsemoyin, Pastor Stephen Kenny, Mr Ben Kanagartnam, Mr & Mrs Sodiq and Rita Ojo, Mr & Mrs Robert and Mado Winn, Dr Faiza Alssaedi, and Dr Samuel Ogundipe. It is good to have you, Dr Samuel, whenever I feel a nudge to discuss research thinking with you. Dr Faiza, you're more than a friend but a sister also.

My high regard to my beautiful wife – Taiwo and wonderful children: Eunice, Dunamis and Treasure, thanks for being my greatest motivation. Taiwo, I really cherish your understanding and selfless effort in raising Dunamis and Treasure together during this pandemic for me to fulfil this coveted dream. The moral support of my parents Mr & Mrs Samuel and Juliana Ajimati and siblings for always calling to check on our wellbeing has made us feel less nostalgic about missing home.

Without you all, it would be lonely planet earth. I pray that every one of you will find a good social network of people to draw strength and support from in private and organisational life, in fulfilling your ambitions, in Jesus Christ Mighty Name. Amen.

Thank you all!

List of Chapter Overview: An Article-Based Thesis

Thesis Chapter	**Communication platform**
Chapter 2 Applying a Multiplex Networks Perspective to Understand Performance in Software Development	This chapter is a paper *accepted* for publication in the *International Journal of Business Information Systems* in 2021. Expected to be finally published early this year 2022. Earlier before being accepted, this was presented as a proposal during a *PhD Poster Presentation at Lero The Irish Software Research Centre Summit*, held in September 2021.
Chapter 3 A Study of Offline and Online Knowledge Brokerage Among Software Developers	This chapter was earlier submitted as a paper to the *Communication of the Association for Information Systems*. It was rejected with good feedback in 2021. It is currently being revised to be submitted to the *International Journal of Information Management* by February 2022.
Chapter 4 The Effect of Advice Network Structure on Problem-Solving Competence Among Software Developers	This chapter is a paper submitted to the *Journal of Systems and Software* and is currently *Under Review* since December 2021.

Chapter 1 Introduction

1.1 Chapter Outline

This chapter provides a detailed overview of the research focus and context to motivate research inquiry and some insights into the summary of findings of this study. Section 1.2 introduces the context and research question. Section 1.3 explains the conceptual background. Section 1.4 highlights the research objectives. Section 1.5 provides information about the research positioning and contributions. Section 1.6 discusses the research design. Finally, section 1.7 provides a synthesis of the findings of this thesis.

1.2 Introduction: The Relational Side of Software Development

Software development requires collaborative knowledge creation and utilisation to achieve performance. This is because to produce software, software teams need to perform tasks related to analyses, designs, development (writing code and configuring), testing, deployment, and maintenance activities. Oftentimes, this demands individual software developers to share knowledge and solve problems together to drive individual and collective decisions in performing these development tasks in knowledge-intensive organisations (Iden & Bygstad, 2018; Rus, Lindvall, & Sinha, 2002; Sawyer, Guinan, & Cooprider, 2010; Šmite, Moe, Šāblis, & Wohlin, 2017). "Software developers" here describe the role of "*programmer, analyst, system tester, database administrator, domain expert, technical leads and project managers*" (Sawyer et al., 2010, p. 82).

To understand how software developers work, paying attention to their knowledge interactions is important. Studies have identified technical and business knowledge interactions as core to software development requirements among software developers. Business domain knowledge is needed to address business processes and customer needs such as analysing requirements. Technical knowledge is needed to solve specific technical problems to develop solutions such as writing and changing source codes to fix bugs

(Çaglayan & Bener, 2016; Rus et al., 2002; Tiwana, 2004, 2012; Vermerris, Mocker, & Van Heck, 2014). Despite the two knowledge interactions being complementary in software development, scholars have argued that they tend to be communicated and studied separately (e.g., Tiwana, 2012; Vermerris et al., 2014). In other words, the lack of clarity surrounding their network structures is problematic in fully accounting for how these knowledge types are sourced and utilised across project functions since such knowledge interactions often occur informally (Iden & Bygstad, 2018; Šmite et al., 2017; Tiwana, 2012; Vermerris et al., 2014). Lin, Chen, Hsu, & Fu (2015) suggest that the understanding of network structure and network knowledge cannot be separated if there were to be an adequate understanding concerning how an individual identifies problems and seek knowledge from others for solutions. In the same vein, other studies suggest that understanding the nature and role of knowledge being shared between individuals can generate insights into meaningful interactions in organisational networks (Moser, Groenewegen, & Ferguson, 2017). Yet, despite the importance and complementary role of these knowledge types in software development knowledge interactions among software developers prior scholarly work on software development in an organisational context fails to consider the direction of these knowledge linkages in their network relationships (Šmite et al., 2017; Vermerris et al., 2014). This includes understanding the role of knowledge and their linkages binding individuals in offline and online ties on job performance (Cross, Borgatti, & Parker, 2001; Moser, Groenewegen, & Ferguson, 2017; Sykes, Venkatesh, & Johnson, 2014; Zhang & Venkatesh, 2013) and leader-follower and follower-leader ties in software development (Peng, Wan, & Woodlock, 2013).

1.2.1 Network Theory as A Theoretical Framework

The usefulness of either business knowledge or technical knowledge cannot be applied without the other to produce software. Current work on software development, however, largely tends to ignore the organisational context of such collaborative efforts, centring on the technical network instead. Driven by social network theory, however, intraorganisational network scholars suggest that analysing the social networks of individuals operating in an organisational context requires considering their knowledge interactions formally and informally as different networks. Such interactions may take effect via various routes, within individuals interacting across offline and online communication networks (Bolander, Satornino, Hughes, & Ferris, 2015; Ceci & Iubatti, 2012; Aalbers et al., 2015). For instance, in informal advice networks, researchers have advocated an inquiry into understanding the dynamics of network ties, especially regarding the type and usefulness of knowledge being exchanged in both offline and online networks (Aalbers & Dolfsma, 2015; Cross et al., 2001; Moser et al., 2017; Reus, Moser, & Groenewegen, 2020; Sykes et al., 2014). This provides a laser focus to investigate business domain and technical knowledge sharing and brokering activity in online and offline communication networks. Thus, it is worth considering how networking and negotiating access for knowledge resources in informal networks shape the development effort that is needed to gain problem-solving competence and task performance. Further studies combining business and technical expertise also reveal its influence on knowledge integration and project team performance (Lee, Park, & Lee, 2015; Mehta & Bharadwaj, 2015) and software development success (Šmite et al., 2017). For example, one online study applied a multiplex network approach to studying informal advice networks and found a positive association between organisations embeddedness in knowledge diversity (business/technical related) and successful innovation (Belso-Martínez, Mas-Tur, & Roig-Tierno, 2017). Still, theoretical and empirical evidence is lacking in relation to the dynamics and interdependence between offline and online communication structures and their knowledge linkages (Moser et al., 2017). To understand network ties and performance consequences in software development settings, hence, it is more valuable to consider the interactions of individuals across the relationship, resource, and communication networks in collaborative knowledge work.

With the intent to provide a theoretical contribution to the literature on intra-organisational collaboration for software development, I examine two distinct networks together and show that there is some sort of complementarity between the two. In such pursuit, this research utilises social network theory to introduce and examine the concept of multiplexity within software development settings. Multiplexity is a useful network perspective in explaining overlapping (two or more) ties and roles in sharing information and knowledge resources between two individuals in different contexts and settings (Lazega & Pattison, 1999; Maruping & Matook, 2020; Verbrugge, 1979). In this context, the application of multiplexity is to provide a conceptual and empirical lens through which an individual organises different network mechanisms to seek diverse knowledge resources in coordinating work activities within their work roles to their performance advantage in informal advice networks. This is to further an understanding of network dynamics regarding the interplay between the role of individuals, knowledge, and the network structure where they are embedded in social networks. Network dynamics refers to changes in network behaviour such as the choice of relationships, ties, roles, or persons in relation to time, events or activities (Chen, Mehra, Tasselli, & Borgatti, 2021; Hernandez & Menon, 2021).

1.2.2 Research Question

The importance of the network of problem-solving ties suggests that when two people share knowledge during problem-solving activities, the resulting performance is beneficial to both of them (Shah, Cross, & Levin, 2018). The broader organisational and software development literature, in this context,   have shown the effect of team problem-solving competence on the overall quality of product and ability to manage requirements uncertainty (e.g., Yuzhu Li, Yang, Klein, & Chen, 2011). Similar studies found that locating and aligning business and technical expert knowledge is positively associated with greater problem-solving competence and overall performance of software development teams (Lin et al., 2015). Despite these significant contributions, there seems to be no good understanding as to whether group members access both business domain knowledge and technical knowledge from the same or different individuals and the potential variation in their problem-solving competence. This is important to understand the extent of communication, collaboration, and sharing of knowledge from multiple sources to solve different problems and achieve greater work efficiency and effectiveness towards software development success. In doing so, further research can help in identifying high and low-performing teams or individuals and their demographic characteristics (e.g., work roles, rank, experience etc.) and how best to manage them more effectively. This thesis aims to resolve these gaps by drawing on social network theory specifically structural hole theory to investigate the relationship between the diversity of network structures and knowledge relations and performances of software developers.

Thus, the research question is: *How does the network structure of individual software developers help to coordinate intra-organisational knowledge resources as a driver to performance in software development?*

1.3 Conceptual Background to A Multiplex View of The Firm: A Review of Literature

To answer the above research question, this thesis draws on network theory as its theoretical anchor, viewing upon the firm as a constellation of social relations in which different networks may be discerned simultaneously (Aalbers & Dolfsma, 2015; Borgatti & Halgin, 2011; Methot & Rosado-solomon, 2019). But what are these networks conceptually, and how do they contribute to knowledge sharing, problem-solving and performance in software development?

According to Schreiber & Zylka (2020, p. 3), social networks are based on "*rational, structural and functional*" dimensions. The *rational* relates to links binding two individuals and are qualified by intensity (relationship strength based on communication), reciprocity (mutual support), and multiplexity (multiple interactions in a tie). The *structural* relates to network properties such as size and density. The *functional* relates to transactional content in terms of information and knowledge moving through a network that explain how two individuals work together. Taken together, the rationale for multiplexity can provide a better picture of the relational strength and level of mutual support between two individuals when they share multiple connections (e.g., diverse knowledge resources) than otherwise with uniplex ties for coordinated action and performance. Table 1 below details different contexts and examples through which network structures can be interpreted to have manifested or considered to be manifesting uniplex and multiplex ties in social networks.

Table 1. Context and Examples of Uniplex versus Multiplex Ties

Uniplex Ties	Examples	Level of Analysis	Multiplex Ties	Examples	Interpretation Reference
Purely offline or online relations only	*Actor A* communicates only face-to-face with *Actor B.*	Individuals	Overlapping relations across offline and online media	*Actor A* communicates through face-to-face/virtual with *Actor C.*	*Haythornthwaite 2001; Mesch & Talmud 2006; Zhang and Venkatesh 2013; Ding et al., 2019*
Purely technical or business knowledge relations only	*Actor A* shares only business knowledge with *Actor B.*	Organisations	Overlapping relations across technical and business knowledge	*Actor A* shares business/technical knowledge with *Actor C*	*Balland, Belso-Martínez, & Morrison, 2016; Belso-Martinez et al., 2017*
Purely formal or informal relations only	*Actor A* maintains only official interactions with *Actor B.*	Individuals	Overlapping relations across formal and informal structures	*Actor A* maintains official/unofficial interactions with *Actor C.*	*Aalbers et al., 2014 ; Cai et al., 2018*
Purely social or economic relations only	*Actor A* maintains only social interactions with *Actor B.*	Organisations	Overlapping relations across social and economic transactions	*Actor A* maintains social/economic interactions with *Actor C.*	*Gimeno & Woo 1996; Ferriani et al., 2012*
Purely task or relationship (friendship) conflicts only	*Actor A* discusses only task issues *with Actor B.*	Individuals	Overlapping relations across task and relationship conflicts	*Actor A* discusses task/personal issues with *Actor B.*	*Methot et al., 2016; Hood et al., 2017; Marineau et al., 2018*
Purely assessing or assisting roles	*Actor A* only evaluates the work role by *Actor B.*	Individuals	Overlapping relations across assessing and assisting roles	*Actor A* evaluates/ supports the work role by *Actor B.*	*Maruping & Matook 2020*
Purely coordinating or gatekeeping or representing or consulting or liaison roles	*Actor A* only coordinates internal knowledge access between *Actor B* & *Actor C.* (*only coordinating role*)	Individuals	Overlapping relations across coordinating, gatekeeping roles etc.	*Actor A* coordinates internal/external knowledge access between *Actor B* & *Actor C.* (*Both coordinating/gatekeeping roles*)	*Gould & Fernandez 1989; Kirkels & Duysters 2010; Aalbers & Dolfsma 2015*

With network theory as its theoretical base, this thesis applies a multiplex network perspective to the firm. This is to help make sense of the inherent (dis)connections between individuals' interactions with diverse networks (e.g., business/technical networks) and performances. Comparatively, a uniplex network has a one-dimensional link while a multiplex network has multidimensional links binding two individuals or organisations in network relationships (see Table 1). This can help understand the potential advantages that multiplex ties may have over uniplex ties in workplace relationships. Addressing the research question is important because studies have found that interpersonal ties in network relationships can reveal much about how individuals coordinate work by collaborating, sharing knowledge and solving problems to derive performance advantages (Peng et al., 2013; Sawyer et al., 2010; Šmite et al., 2017). Particularly in an intra-organisational context, organisation scholars have been keen to better understand network pluralism, the coexistence of various networks simultaneously (Liu, Park, Hymer, & Thatcher, 2019; Methot & Rosado-solomon, 2019; Shipilov, Gulati, Kilduff, Li, & Tsai, 2014).

To unravel how the layering of networks may contribute to knowledge sharing, problem-solving and performance in software development in this thesis, I additionally draw on structural hole theory, which is applied to understand the role of individuals and holes in multiple network structures on resources access and work performance. Structural hole theory seeks to understand the benefits individuals accrue when they serve as brokers or bridges in a network in connecting otherwise disconnected others with knowledge and informational resources within and between network structures (Belso-Martínez, Mas-Tur, & Roig-Tierno, 2017; Burt, 2004; Obstfeld, 2005; Sasidharan, Santhanam, Brass, & Sambamurthy, 2012). A structural hole theory is chosen instead of a gatekeeper theory because the former provides more understanding about network brokerage than the latter, which is more focused on scouting and filtering knowledge from external sources to be shared with two members internally to perform tasks. Studies suggest that not all individuals are often directly connected to the relevant knowledge sources to access new and not redundant knowledge without passing through brokers to provide solutions to work problems (Belso-Martínez et al., 2017; Cataldo & Herbsleb, 2008; Whelan, Teigland, Donnellan, & Golden, 2010). The rationale for applying structural hole theory here is to understand the way network brokerage alter performance-related benefits (e.g., problem-solving competence) in multiplex networks (business/technical networks combined) of software developers.

Multiplexity and Software Development

In the review of the intra-organisational network literature as applied to a software development context, there are studies to show that the software development process requires major informal networking for knowledge resources between software developers (Boden & Avram, 2009; Sasidharan et al., 2012; Šmite et al., 2017). Krackhardt & Hanson (1993) claim that much work in organisations occurs through informal networks as they help employees to bypass "*formal reporting procedures to jump-start stalled initiatives and meet extraordinary deadlines*" (Krackhardt & Hanson, 1993, p. 104). Informal networks refer to networks of interactions between individuals that are not officially charged or mandated to work together on a given work role and project. These networks have been found to provide greater flexibility in accessing vital information and knowledge resources that can address unanticipated problems in formal work routines (Aalbers, Dolfsma, & Koppius, 2014; Ben-Menahem, Von Krogh, Erden, & Schneider, 2016; Brennecke & Rank, 2016; Brown & Duguid, 1998; Krackhardt & Hanson, 1993). Still, researchers are seeking clarity as to how a software developer interaction influences different network outcomes in the software development environment (Iden & Bygstad, 2018; Peng et al., 2013; Sawyer et al., 2010; Šmite et al., 2017). Beyond the software domain, studies suggest the need for more research into the role of intraorganisational networks (i.e., internal organisation structures) on knowledge sharing and innovation development and performance (Aalbers, Dolfsma, & Koppius, 2013; Tassabehji, Mishra, & Dominguez-Péry, 2019). In this context, informal networks refer to the business and technical knowledge networks through which business and technical knowledge are shared inside an organisation. These are generally termed advice networks. These words are used interchangeably in this thesis.

In view of the foregoing, there are different research debates, particularly in multiplex-based studies seeking to understand the interaction between network structures and performance. The research on communication network perspective found that individual embeddedness in offline and online networks provides opportunities to access and utilise diverse knowledge to their advantage (Ding, Bhattacharya, Phan, & Kong, 2019; Zhang & Venkatesh, 2013; Zhang, Venkatesh, & Huang, 2008). For example, Zhang & Venkatesh (2013) found that embeddedness in offline and online networks resulted in greater knowledge resources and performance. From the resource network perspective, studies revealed that individuals embedding in workflow and software advice networks are better performers (Sykes et al., 2014). Furthermore, based on the relationship perspective, studies beyond software development argue that individuals networking across formal and informal work structures are more innovative and better performers (e.g., Aalbers et al., 2014; Cai et al., 2018). In essence, understanding the interactions of individuals in multiplex networks would generate more insights into the performance of software developers (Schreiber & Zylka, 2020). It is argued that multiplexity offers a multilayer and accurate account of network structures (Bliemel, McCarthy, & Maine, 2014; Filiposka, Gajduk, Dimitrova, & Kocarev, 2017; Lee & Lee, 2015). For example, in an ideal world in the software development process, two software developers can but not necessarily engage in exchanging business domain and technical knowledge simultaneously. Likewise, in sharing either or both knowledge types through face-to-face offline and virtual online collaboration to perform individual and collective development tasks. Therefore, the dynamic interaction between the network structure and knowledge of software developers requires a multiplex network perspective to generate better insights on their implications for performance than their uniplex evaluation. This includes understanding the structure of software developers that constitute high performance (Peng et al., 2013). Filiposka *et al.* (2017) suggest that multiplexity tends to provide more depth in understanding and interpreting the difference in separate, interrelated, and overlapping connections in social networks. This thesis takes a more granular and integrated approach via multiplexity to study software developer networks, in particular the nature of knowledge sharing and communication roles (e.g., gatekeeper, coordinator) in informal advice networks of software developers. This is to provide a better understanding of what the social network of software developers means for their knowledge sharing, problem-solving and performance.

1.4 Research Objectives

As outlined via the above research question in section (1.2), the purpose of this thesis is to examine the effect of multiplexity on the network structure and performance of software developers. Given that the role of network structure, knowledge and actors are in tandem in social network interactions for achieving performance, this thesis considered the following objectives.

The **first objective** focuses on understanding network interactions of the business domain and technical knowledge resources in offline and online communication networks of software developers. It is thus important to understand how the type and extent of knowledge flowing and binding two individuals together in offline and online communication structures become organised simultaneously and vary potential network benefits. Each knowledge serves a different but shared meaningful purpose and may not necessarily be located and accessed from the same or different individuals. Thus, the need to examine their network structures and how they are coordinated to drive performance goals. In this vein, Schreiber & Zylka (2020) found that the lack of a broader multiplex (but uniplex) network perspective does not help to capture how different knowledge interactions overlap and shape performance in software development settings. A similar argument exists within the context of entrepreneurs and their performance advantages (Bliemel et al., 2014), as well as in understanding the difference in creative interactions of an individual in intraorganisational networks (Aalbers et al., 2014; Lee & Lee, 2015) or firm innovativeness in inter-organisational networks (Giovanna, Rojas, Rogelio, Solis, & Jianjun, 2018). By means of this first research objective, this thesis explores the concept of multiplexity, as a broader network approach in social networks to understand how it has been applied in software development settings using a systematic literature review and to drive an agenda about its application. The study presented in chapter 2 (article 1) thereby focuses on this conceptual framing of multiplexity in network-based studies.

The **second objective** focuses on understanding the impact of brokerage roles in software development knowledge sharing interactions and the development effort of software developers in multiplex networks. In other words, it seeks to understand the relationship between diverse brokerage roles (e.g., gatekeeper, coordinator, etc.) through network linkages and knowledge resourcefulness of software developers. This is built on the dimensions of multiplexity identified in the first objective. The study presented in chapter 3 (article 2) thus examines empirically the dynamics of brokerage roles in the business domain and technical knowledge sharing across offline and online communication networks on problem-solving potentials in software development.

The **third objective** focuses on understanding the interacting impact of brokerage and contact quality in multiplex networks on the problem-solving competence of software developers. The study presented in chapter 4 (article 3) thereby explores the interaction of brokerage in the business domain and technical networks with a focus on contact quality on the problem-solving competence of software developers. This is to understand how these different informal network structures shape the ability of software developers to develop or increase the capability to solve a variety of problems more effectively (see Table 2). Organisational network scholars investigating multiplexity for instance typically build on structural reasoning and assume that the existence of one type of relation fosters the creation of other types of relations (Borgatti & Cross, 2003; Rank & Tuschke, 2010). By means of this objective, this thesis explores the coexistence of various intra-organisational networks between software development stakeholders, attributing the relational effect to the individual actor's embeddedness in either of the networks – the business and the technical network. In sum, the motivation is to understand the causality undergirding both business and technical networks in software development knowledge sharing in offline and online collaborations and how this varied performance in the software development effort.

Overall, objectives 1-3 seek the application of a multiplex lens in approaching and appreciating the study of software development networks from knowledge and communication diversity (multilayer interactions) than mere simplicity of single level interactions to understand performance differentials in software development settings.

This thesis explores the nature of multiplexity in terms of an integral view of the business domain and technical knowledge sharing in offline and online communication on the problem-solving competence of software developers. Specifically, these areas of literature guide further these questions below.

Q1. *How can a multiplex-based view contribute to software developers' coordination of resources towards performing software development tasks and optimising greater work outcomes? (Article 1)*

Q2. *How do online and offline knowledge brokering activities contribute to software development? (Article 2)*

Q3. *What relationship between business and technical advice networks and their brokerage activities affect software developer problem-solving competence in software development? (Article 3)*

The conceptual framework in answering the three research objectives and questions is proposed in this chapter 1 under section 1.5 in figure 1.

Table 2. Summary of Characteristics of Articles included in the Thesis

	Article 1	Article 2	Article 3
Title	Applying a multiplex network perspective to understand performance in software development	A study of offline and online knowledge brokerage among software developers	The effect of advice network structure on problem-solving competence of software developers
Publication outlet	International Journal of Business Information Systems (***Awaiting publication – i.e., in production***)	To be submitted to International Journal of Information Management	Submitted to Journal of Systems and Software ***(To be revised and resubmitted))***
Author	Ajimati Matthew Oladeji Aalbers Hendrik L Whelan Eoin van Kranenburg Hans L Morgan Lorraine	Ajimati Matthew Oladeji Whelan Eoin Aalbers Hendrik L Morgan Lorraine van Kranenburg Hans L	Ajimati Matthew Oladeji Whelan Eoin Aalbers Hendrik L Morgan Lorraine van Kranenburg Hans L
Research Objective	The paper aims to **1)** review and conceptualise the existing approach to multiplex social network phenomenon and **2)** then drive a research agenda for its relevance in software development settings.	The paper aims to **1)** empirically explain the impact of multiplex networks in software development knowledge sharing ties. **2)** guide theoretical and managerial implications in managing software developer networks.	This paper aims to **1)** conceptually and empirically assess the effect of multiplex relationships and individual effort in developing problem-solving competence. 2) guide theoretical and managerial implications in facilitating work effectiveness and project success.

Theoretical framework	Social network literature	Social network and brokerage literature	Social network, brokerage and problem-solving literature
Research Method	This study conducts a systematic literature review and reviews 163 articles published in management science, information systems, and software engineering field, 1990-2020.	This study uses a cross-sectional, explorative, and descriptive method to generate insights from software developers.	This study uses a cross-sectional and descriptive approach to generate insights from software developers.
Data Source	Extant empirical and conceptual studies	Data consists of 6 interviews (pilot), 25 sociometric survey responses from software developers	Data consists of 153 sociometric survey responses from software developers
Findings	Integrated conceptual application of multiplexity in software development	*Offline brokerage*: The presence of communication roles in the business domain/technical knowledge sharing. *Online brokerage*: The absence of any communication roles in business knowledge sharing. But the presence of only coordinator/representative roles in technical knowledge sharing	A positive effect of brokerage in a business domain/technical advice connectedness on problem-solving competence. The positive effect of business advice connectedness on problem-solving competence (*moderating effect of contact quality*). The negative effect of the direct impact of technical advice connectedness on problem-solving

			competence (*moderating effect of contact quality*). The negative effect of contact quality (moderating business/technical advice networks) on problem-solving competence.
Theoretical Contributions	Proposed an integrated framework of multiplexity and application in software development knowledge sharing processes and what it means for performance.	Provides an understanding of brokerage/network dynamics in software development knowledge sharing/problem-solving potentials.	Provides an understanding of network effect (esp. brokerage) on problem-solving competence
Contributions to Practice *(Managerial)*	Promotes interpersonal diversity for diverse knowledge flow in workplace relationships	Promotes brokers (or experts) interventions and balanced brokerage activities in knowledge resourcefulness/linkages in offline & online collaborations in the software development effort	Promotes the need for developing problem-solving competence to drive software development success
Literature Support – *in research calls*	*Schreiber & Zylka, 2020*	Aalbers & Dolfma, 2015; *Šmite, Moe, Šāblis, & Wohlin, 2017; Moser, Groenewegen, Ferguson, 2017; Reus, Moser & Groenewegen, 2020)*	*Lin, Chen, Hsu & Fu, 2015*

1.5 Research Positioning and Contributions

This thesis addresses issues of multiplexity, brokerage, and problem-solving competence of software developers. These areas serve as a theoretical lens to examine and manage the social process of software development. The "*social process*" is defined by how individual software developers interact by working together to develop software. These patterns of interactions have been found to shape performance in software development (Sawyer et al., 2010). As earlier noted, software development is common to software engineering and information systems fields because of its focus on the production of software products for business use by customers and end-users. A big part of software development is the need for software developers to collaborate, communicate and coordinate activities together. This is often achieved by sharing knowledge with people across many functions that cut across intrateam and inter-team work activities in tackling development problems (Iden & Bygstad, 2018; Rus et al., 2002; Sawyer et al., 2010).

By focusing on the knowledge and human side of the problem-solving activity in software development, this thesis considers multiplexity as a proxy to understand the coordination of network relations pertinent for knowledge resourcefulness in driving higher software development performance. Because multiplex networks have different structural mechanisms and communication sources through which two software developers may be diversely connected or linked to discuss and coordinate information and knowledge resources to perform development activities. Therefore, it is important to understand how much access and use of knowledge people can control in work-related activities (Cheng, 2009). This is because different avenues can be efficient for sharing knowledge resources and when well managed can influence greater performance (Ashworth & Carley, 2006; Cross et al., 2001; Du, Ai, & Ren, 2007; Hossain & Zhu, 2009). Thus, the diversity and multilevel interactions in intraorganisational informal advice networks driving performance-related outcomes are key, which studies suggest have not been fully explored in software development research (e.g., Schreiber & Zylka,

2020) due to a dominant uniplex network perspective. For example, a uniplex network perspective, which reveals that technical knowledge ties contribute to the greater software development effort (Cataldo & Herbsleb, 2008) tends to offer limited insights into the structure of software developers on software development performance.

Existing studies provide disjointed accounts of network orientations as follows. A resource orientation shows that individuals' effort in brokering (e.g., gatekeeping, coordinating etc.) for business domain and technical knowledge sharing has a positive effect on knowledge integration and performance of software developers (Mehta & Bharadwaj, 2015; Sawyer et al., 2010; Šmite et al., 2017). Yet other multiplex studies with a deterministic communicational view suggest that individuals need to be interlocked in offline and online networks to benefit from diverse knowledge and performance (Zhang & Venkatesh, 2013). Other proponents of a structural relationship perspective conceive that neither resource nor communicational orientation matter without individuals being linked by both formal and informal relations to generate creative knowledge and amplify performance (Aalbers et al., 2014; Cai et al., 2018). Although these are valid contributions to managing knowledge flows in boundary-spanning activities, such contributions are inconclusive because individuals may share joint orientations strategically based on networking in multiple network structures to acquire and manage diverse knowledge resources to their advantage. In doing so I add to current relational pluralism work (e.g., Shipilov et al., 2014) as it can help to understand the process by which individuals build communication, trust, drive efficiency, achieve resource and network stability, and work interdependently; which literature beyond software development settings suggest are possible in network multiplexity than one-dimensional relations such as uniplex ties (Lazega & Pattison, 1999; Min, Yi, Lee, & Goh, 2014; Shipilov, 2012; Zerbini & Castaldo, 2007).

Above all, the application of multiplexity, therefore, brings to the fore integrated dimensions and dynamics of network ties (interactions), which a uniplex network perspective may not discern, or capture the different angles to such interpersonal connectedness in social network structures in software development. Such multiplex network orientations involve cross interactions of offline and online communication ties, formal and informal ties, business and technical (including friendship) ties that can be configured as a network that combines and utilises different access points – a coordinating mechanism for generating knowledge resources to aid better performance. In a nutshell, multiplex network structures generate deeper insights about possible variations in online and offline network behaviours in the business domain and technical knowledge brokerage (e.g., gatekeeping, coordinating etc.) and strength of ties (relational bond) between software developers. All these can reveal much about how software developers exhibit greater network potentials, for example, in developing the required effort to gain problem-solving competence and its significance in managing software development teams and practices. Therefore, and more importantly, while this study is positioned within a multiplex discourse in understanding software development networks for performance, the approach applies to other networks of knowledge workers in any work environment within the field of management, information systems, and software engineering. The conceptual model for this thesis (see Figure 1 below) revealed the relationship of the interacting elements of all articles combined in studying and managing individuals and teams in software development activities.

The contributions of this thesis are twofold: First, it improves understanding of the social network of multiplexity and brokerage for performance in software development settings. Second, it contributes to the management of software development teams related to the structure, knowledge, and individual brokerage orientation in coordinating development activities. In sum, this study contributes to the social network research relating to multiplexity, brokerage, and problem-solving competence to enrich insights into the management of software teams to drive better performance among the academic and practitioner community.

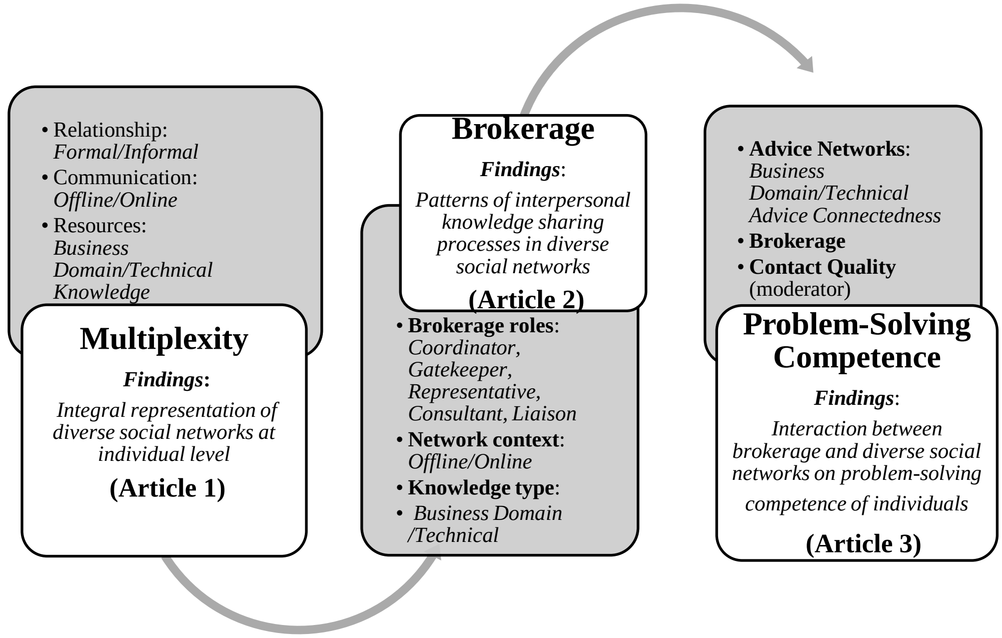


Figure 1. Conceptual model of this thesis

In sum, article 1 findings were based on the different dimensions that social network ties entails, here refers as multiplexity while article 2 examines the role of brokers and the communication roles they occupied or not occupied within and across multiplexity dimensions and article 3 considers the relationship between these resource dimensions of advice network multiplexity and effect on individual problem-solving competence in driving successful software development effort.

1.6 Research Design

A research design has been described as a logical plan and selection approach to a research question (problem phenomenon) by which the researcher collects and analyses the relevant data based on a research paradigm to drive the research purpose (Blanche, Blanche, Durrheim, & Painter, 2006; Creswell, 2009; Groenewald, 2004). A research paradigm refers to certain ways of thought in undertaking research studies and are often guided by specific assumptions, ideas, and values of scientific and academic traditions. A research paradigm typically includes the ontology, epistemology, and methodology (including methods) guiding the research problem. Thus serving as the underpinning philosophical perspectives as well as the research design used in examining and positioning this research study (Antwi & Kasim, 2015; Blanche et al., 2006). According to Blanche et al. (2006), the research questions, methods and paradigms must fit together as a process of implementing the research. Hence, for the benefit of linking the proposed research problem to philosophical perspectives, this thesis seeks to examine (i) how software developers explore, navigate, and exploit the diversity of multiple knowledge resources and knowledge sources to perform software development tasks and (ii) how the different knowledge-sharing resources and knowledge sources are combined and utilised to their advantage, particularly in developing the individual effort required to increase their problem-solving competence. In other words, this research seeks to understand how software developers consolidate and benefit from business domain and technical knowledge sharing from multiple sources in their (offline[*] and online communication included) advice networks and its network performance-related advantages. Such as in improving software development effort vis-à-vis knowledge sharing and problem-solving that are pertinent in driving successful software development.

The objectives driven by this research problem will form the premise for this research philosophy. It is assumed that the nature of the research question drives the philosophical rationale that may be adopted (Saunders, Lewis, & Thornhill, 2009). Arguably, any research is taking a philosophical action in the process, as questions under study are philosophical by nature, and thus requires a clear understanding to generate sufficient insights and quality work. The philosophical lens enables researchers to understand situational contexts by formulating research questions, designing methodologies to generate results or findings, and positioning studies within research discourses (Hassan, Mingers, & Stahl, 2018). In this case, multiplexity. The philosophical

[*] Although communication for the past year since COVID-19 has been online, this study successfully gathered data on both communication sources prior to pandemic disruption.

details regarding the various layers of this research design as per articles 1-3 are as follows.

1.6.1 SNA Philosophical Perspectives in this Research Design

To begin with, philosophy is a way to clearly and easily reflect on knowledge statements by carefully analysing them for sense-making and scholarship (Hasan, 2016). Considering that different fields have different schools of thought guiding them, research philosophers believe there is no one-size-fits-all solution to any problem or a way of understanding the social world. But then, such a problem may deserve a simplistic approach to shedding light on it by generating a specific or some general understanding of the social phenomena (Hassan et al., 2018).

Bhaskar (1975) suggests that critical realism considers the domain of the *real*, the *actual* and the *empirical,* in making sense of the social systems. The domain of the real is externally oriented and is characterised by structural mechanisms with causal properties and powers that cannot be fully studied. The actual is linked to the changes that are triggered by the exercise or unexercised of power and structures in generating events in the social systems. The empirical is associated with the capturing of small parts of these events by way of observation and record-keeping by the observer or researcher in driving scientific research. Thus, it is argued that systemic thinking based on the empirical and experiential view of the world provides deeper knowledge of the truth concerning "*one real"* world. In other words, this view share beliefs that to understand causal explanation and generative causality regarding worldviews; it needs to seek knowledge surrounding regular and irregular, observed and unobserved, perceived, and experienced patterns of events, that can but not necessarily be concept and activity-dependent, which may further invite explanation and interpretation of the human meaning associated with it. Critical realist posits that human have causal powers to drive changes in social structures, but their understanding of the world is not limited by their conceptualisation of its existence. Roy Bhaskar argued that attempting to do so would be committing an epistemic fallacy against the ontological stratification in critical realism (Bhaskar, 1975; Mingers, 2000; Mingers

& Standing, 2017). An epistemic fallacy would be to reduce the understanding of the world or events to what is strictly controlled or identified, to be regularly and directly observable, without seeking answers for the unobserved causes that may deserve other human interpretation.

This thesis adheres to the relative view of social science based on the critical realism paradigm, which suggests the mixed-method research design, where qualitative interviews and quantitative surveys are considered as balanced approaches to generate sufficient useful insights. Specifically, one of the articles (article 1) was purely designed using a systematic review to conduct an exploratory study of the literature by reviewing 163 articles (consisting of empirical and conceptual studies) on social network studies in the software development context. This has led to understanding and establishing the different dimensions associated with the application of multiplexity in the extant studies, and as such, posit an integrated framework relevant for social network analysis and interpretation. Article 2 was designed using a sociometric (network) survey instrument with a dataset of 25 software developers. However, pilot interviews with 6 software developers were conducted and stored in NVivo software to generate the relevant knowledge interactions for survey measures, which was analysed using UCINET 6.0 organisational analytic (statistical) tool. The empirical insights generated captured dynamism in software development knowledge sharing and variations in communication roles across offline and online networks. Article 3 was also designed using the same survey instrument to collect data from 153 software developers, which were analysed using Partial Least Square Structural Equation Modelling (PLS-SEM 3) to draw causal explanation and interpretation regarding the benefit of multiplex advice networks on individual problem-solving competence of software developers.

Above all, in understanding the epistemological thinking of multiplexity and the synergy of its dimensions in resource, communication and relationship orientations, its understanding would deserve applying a blend of multiple methodological approaches. At the nexus of multiplexity is multiple dimensions of social ties in social structures, which accounts for the roles of individuals, attributes, network positions, knowledge interactions, mode of communications and software development activities. All of which contributes to the dynamics and relevance of multiplex reality. Interestingly, Whelan et al. (2016, p. 7) posit that "*Social ties can be thought of as either objective phenomena (something that has defined features) or subjective (something that has features only to the degree that they are accepted/understood/constructed by the participants*". In other words, in understanding the meaning associated with the reality of a multiplex phenomenon, combining, or embedding both quantitative and qualitative research design can deepen understanding of the essence of its reality. This choice has informed the research design of this thesis. Even in further advocating a shared view that can accommodate diverse methodologies in approaching and understanding multiplexity.

1.7 Synthesis of Findings of Article Related Chapters

The main purpose of this thesis is to understand, explain and extend the knowledge importance of the diversity of social network interactions in knowledge sharing processes, problem-solving activities, and performance of software developers. This thesis has established that software development is both knowledge and people-intensive because to produce software products, software developers would need to interact with their colleagues or co-developers to tap from their expert knowledge of know-how and know-why in solving problems to completing software development tasks. This thesis thereby pays attention to the structure of software developers (i.e., software developer networks) in sourcing and combining business and technical knowledge resources to manage software tasks and achieve performance advantages.

This thesis via article 1 adopted a systematic literature review from a multidisciplinary stream of literature to show that multiplexity – which deals with diverse and overlapping relations of network ties or connectedness – can cut across formal and informal work structures, offline and offline communication structures, business, and technical network structures. In other words, this exploratory study of social networks revealed that intra- and inter-organisational network ties need not be purely formal, informal, offline, online, business, or technical interactions, but can also overlap within the same network of relationships. The systematic review suggests that when relations involving two people or organisations have one dimension, such relations are uniplex and thus represent a one-mode network. Whereas overlapping relations of different network structures suggest a multidimensional orientation called multiplex and thus represent a two-mode network.

The main findings of this article 1 is that multiplex ties can provide more explanatory power of the importance of network diversity than uniplex ties in understanding an individual's coordination patterns and how they shape performance differentials among software development teams. While the extant studies revealed multiplex applications and performance differences within and between organisations, this thesis considers its application between individuals in software development teams. This premise forms the foundations for conducting empirical studies and drawing inferences concerning the relationship between different knowledge resources, including offline and online sources of brokerage in article 2, and how brokerage in diverse advice networks, namely business and technical advice connectedness for knowledge resources, influence individual problem-solving competence of software developers in article 3.

Building on the earlier diversity of network ties, article 2 empirically examined offline and online business knowledge brokerage as well as the offline and online technical knowledge brokerage in software development knowledge sharing processes. Brokerage in terms of the functional roles of coordinator, gatekeeper (internally oriented), representative, consultant, and liaison or itinerant (externally oriented). The thesis findings here revealed more knowledge brokering activities in the offline and online technical knowledge sharing than the offline and online business knowledge sharing. Specifically, the lack of communication roles and brokers in the online business knowledge sharing and two missing roles in the online technical knowledge sharing demonstrate significant gaps in software development knowledge processes that can affect the ability of software developers to source, expand, distribute, and integrate different knowledge resources in the software development effort.

Furthermore, article 3 builds on the advice networks that software developers maintain in sourcing for business domain and technical knowledge resources and their effect in shaping their problem-solving competence. In doing so, it theorised that business advice and technical advice connectedness and a brokerage in business advice and technical advice connectedness for knowledge resources would enhance the problem-solving competence of software developers. But that the effect of business advice connectedness and technical advice connectedness on problem-solving competence depends on the quality of contacts within these networks. The finding revealed that software developers who engage in knowledge brokering in both business and technical advice (network) connectedness will particularly increase problem-solving competence to drive their software development effort. However, this finding also revealed that contact quality played no significant moderating role between business and technical advice connectedness and problem-solving competence.

Chapter 2 Applying a Multiplex Network Perspective to Understand Performance in Software Development

2.1 Introduction and Chapter Outline

This chapter sets the stage for understanding the conceptual lens and application of multiplexity in social networks of software developers as a plausible research agenda in software development settings in this thesis. Section 2.2 provides an abstract of this study. Section 2.3 introduces the backdrop of this research. Section 2.4 explains the methodological approach in conducting this study. Section 2.5 reviews the relevant literature in this context. Section 2.6 provides a discussion of the results. Lastly, section 2.7 describes the conclusion of this research.

2.2 Abstract

A number of studies have applied social network analysis (SNA) to show that the patterns of social interaction between software developers explain important organisational outcomes. However, these insights are based on a single network relation (i.e., uniplex social ties) between software developers and do not consider the multiple network relations (i.e., multiplex social ties) that truly exist among project members. This study reassesses the understanding of software developer networks and what it means for performance in software development settings. A systematic review of SNA studies between 1990 and 2020 across 6 digital libraries within the IS and management science domain was conducted. The central contributions of this paper are an in-depth overview of SNA studies to date and the establishment of a research agenda to advance our knowledge of the concept of multiplexity on how a multiplex perspective can contribute to a software developer's coordination of tasks and performance advantages.

Keywords: multiplexity; software development; communication network; social network analysis; software developer; performance

2.3 Introduction

The purpose of this paper is to propose a research agenda detailing how to multiplex social network research and advance performance in software development settings. Understanding the factors associated with enhanced performance has long been a core theme in software development research (Licorish & MacDonell, 2014; Robillard & Robillard, 2000), and in knowledge-intensive environments in general (Aalbers et al., 2014; Lazega & Pattison, 1999; Lee & Lee, 2015). Within this body of work, a number of studies have revealed that the patterns of social interactions between software developers explain important organisational outcomes (Ghobadi, 2015; Jiang et al., 2019; Licorish & MacDonell, 2014; Park & Lee, 2014; Sawyer et al., 2010; Ulhas et al., 2016). Studies in this tradition apply social network analysis (SNA) techniques to assess how the network positions (e.g., centrality) of software developers vis-à-vis other developers influence access to diverse information and knowledge resources, which in turn influences team cohesion, productivity and quality performance outcomes (Datta, 2018; Ehrlich & Cataldo, 2014; Helms & van Reijsen, 2008; Nan & Kumar, 2013; Nootjarat et al., 2015; Sawyer et al., 2010; Singh, Tan, & Mookerjee, 2011; Yang & Tang, 2004). Studies examine also the effect of the centrality of information versus influence on the variations of software defects and work quality (Datta, 2019). The medium of interaction is also important as the interlocking of offline (i.e., face-to-face) and online (i.e., internet/web) personal contacts are significant in explaining a more accurate representation of collaborative relationships (Gloor et al., 2007) and variances in the individual job (Zhang & Venkatesh, 2013) and student (Ding et al., 2019; Zhang et al., 2008) performance. Since the medium and extent of communication among software developers can determine the success or failure of software projects (Bernardi et al., 2018; Cataldo & Herbsleb, 2008; Defranco & Laplante, 2017; Storey et al., 2017).

Beyond software development settings, network positions have been linked also to innovation performance (Belso-Martínez et al., 2017), innovative knowledge transfer (Aalbers et al., 2014), innovative productivity (Gómez-Solórzano et al., 2019), idea generation (Burt, 2004; Kijkuit & van den Ende, 2010), and overall performance (Cross & Cummings, 2004; Maurer et al., 2011). Indeed, the structural position of an individual both enables and constrains their actions, influencing the resources available to them (Parker et al. 2015). For these reasons, it is important to understand how network structures impact performance in ISD settings.

Network embeddedness matters and such embedding may well take place across various layers of interaction simultaneously (Aalbers et al., 2014; Lee & Lee, 2015; Snijders et al., 2013). However, the existing software development studies which examine network position are limited in that they are restricted to single-layer networks (i.e., uniplex social ties). Within the broader field of SNA, it is now accepted that multilayer networks (i.e., multiplex social ties) provide a much richer understanding of social network dynamics (Bliemel et al., 2014; Ferriani et al., 2012; Lee & Monge, 2011; Shipilov, 2012; Verbrugge, 1979). As illustrated in figure 2 below, uniplex social ties describe how individuals are affiliated by only one resource relationship and membership. For example, in developing a framework to assign people to software development teams, Latorre and Suárez (2017) use social skills as the uniplex tie connecting people. Multiplex social ties, on the other hand, describe the multiplexity of overlapping networks that capture the different types of connections between individuals within the network. Other scholars consider the nature of uniplex social ties to be homogenous based on its single link type, and multiplex social ties as heterogeneous based on its multiple link types, which are more functionally complementary (Meng et al., 2016). For example, studies show that contacts in formal, informal and especially multiplex networks explain transfer of innovative knowledge (Aalbers et al., 2014) and performance (Cai et al., 2018) in an organisation. Multiplex benefits are linked to effective communication, cooperation and strategic collaboration (Haythornthwaite, 2001; Lazega & Pattison, 1999; Maggioni et al., 2013; Pavan, 2014; Shipilov, 2012).

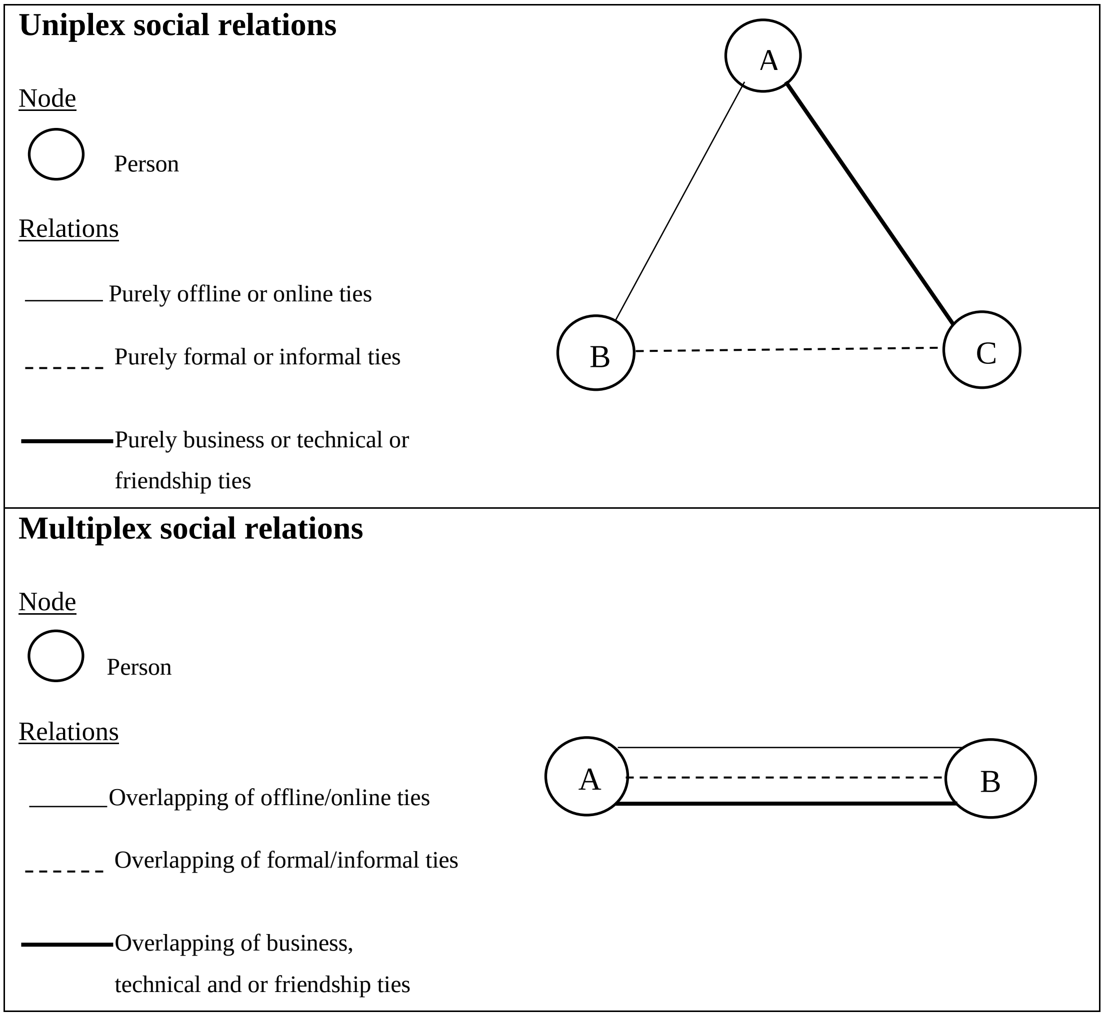


Figure 2. Comparing Uniplex to Multiplex Network

Figure 2 provides a simplified view of the Uniplex versus Multiplex network. The basic difference between the two is that the former shows individual persons being joined by a single link tie while the latter shows relations consisting of different multiple link ties (i.e., holding all elements of ties).

Software developers do not rely on their network of contacts for one type of advice, information, or knowledge. They seek and share a wide diversity of information and knowledge resources: problem-solving, technical developments, decision making, career advice, friendship, social support, and many others. While recognising the reality that multiple types of interactions manifest in social network structures (Nidumolu & Knotts, 1998), researchers across a number of fields call for research work on the effect of multiplex ties on knowledge creation and diffusion in knowledge-based projects (Aalbers et al., 2014; Ferriani et al., 2012; Lee & Monge, 2011; Maggioni et al., 2013). Indeed, recent work on intraorganisational dynamics has started to approach organisations as complex systems in which different networks may be discerned (Aalbers et al., 2013; Borgatti & Halgin, 2011). The basic idea is derived from the observation that people tend to combine different possible aspects of a relationship into a single tie with a concrete other (Mcpherson et al., 2001). To understand performance in software development it, therefore, is important to realise that a person operating in one network can simultaneously be nested in other networks of a different nature. Thus, the purpose of this paper is to propose a research agenda detailing how to multiplex social network research and advance performance in software development settings. Specifically, this study reviewed the multiplexity literature and consider its relevance to software development before posing a series of research questions which future studies should address.

2.4 Methodology

Scholars across management science (Snyder, 2019), information systems (Okoli & Schabram, 2010; vom Brocke et al., 2015) and software engineering (Kitchenham & Charters, 2007) fields have considered literature reviews, especially systematic reviews as a way to structurally present an overview of the research problem and trends without bias in a reproducible manner to identify research gaps, analyse and organise research results, and to drive research agendas towards making contributions to the body of knowledge in academia and industry. As guided by these authors, this study adopted a systematic literature review approach to synthesise extant studies on multiplex approach, which is needed to understand various dimensions (e.g., medium of communication, types of resources shared, and nature of work relationships) of ties inherent in software developer networks that is yet to be explored prior, to fully understand and explain variances in software developer performance.

2.4.1 Research purpose, question, and objectives

The purpose of this paper is to propose a research agenda based on the concept of network multiplexity or multiplex network perspective, to explore how the nature of network embeddedness can better explain performance in software development settings. Recent research tends to have an inaccurate or incomplete view of the embedded nature of software developer networks, which is problematic in understanding their social network structures and performance consequences (Peng et al., 2013; Schreiber & Zylka, 2020). Similar research shows that the characteristics or traits of software developers to network with others for resources access can alter performance differently (Acuña et al., 2009; Licorish & MacDonell, 2014). This includes how software developers coordinate tasks to drive performance. Thus, this study explored a central research question followed by the set of objectives below:

RQ. How can a multiplex-based view contribute to software developers' coordination of resources towards performing software development tasks and optimising greater work outcomes?

RO1. Investigate and explain how a multiplex network perspective can enhance performance in software development.

RO2. Understand how the nature of tasks and individual traits (e.g., personality) shape multiplex ties in software development.

To answer the RQ, this study reflected on RO1 and RO2. Since a multiplex network perspective is not widely popular in software development research but has gained popularity in management science areas, this influenced the current merging and consideration of these different kinds of literature. In this study, we focus on software development performance as outcomes related and social network analysis as methods to evaluate, interpret, and integrate our view of network embeddedness in this research agenda.

2.4.2 Search string

This study developed some search terms based on the scope of the research question and purpose. The keywords were defined based on the initial search of the literature, namely: *multiplexity, multiplex ties, multiple ties, overlapping ties, multiplex relationships; social network analysis, social networks; offline networks, online networks, online communication networks, offline communication networks; software development, information systems development.* These keywords were organised into search strings using two Boolean "OR" and "AND" operators. The Boolean OR operator was used to combine similar terms while the AND operator combine different search terms as follows:

Table 3. List of search strings

1	("multiplexity" OR "multiplex ties" OR "multiple ties" OR "overlapping ties" OR "multiplex relationships" OR multiplex networks") AND ("software development" OR "information systems development")
2	("social network analysis" OR "social networks") AND ("software development" OR "information systems development")
3.	("offline networks" OR "online networks" OR "online communication networks" OR offline communication networks") AND ("software development" OR "information systems development")

2.4.3 Data source

The search strings were conducted across different journal databases. These include Engineering Village (i.e., Compendex), AIS eLibrary, Science Direct, EBSCOhost, Web of Science and Scopus. This was done to gain coverage of the pertinent articles relevant to the scope of this research across information, management, and engineering-related fields.

Table 4. Literature search per selected database

Search string	AIS eLibrary		Science Direct		EBSCOhost		Web of Science		Engineering Village (Compendex)		Scopus	
	Hits	Revi-ewed	Hits	Revi-ewed	Hits	Revi-ewed	Hits	Revi-ewed	Hits	Revi-ewed	Hits	Revi-ewed
("multiplexity" OR "multiplex ties" OR "multiple ties" OR "overlapping ties" OR "multiplex relationships" OR multiplex networks") AND ("software development" OR "information systems development")	3102	125	1258	181	167	29	4007	387	7365	53	455	50
“social network analysis” OR “social networks”) AND (“software development” OR “information systems development”)	5163	107	10425	264	1219	22	652	74	1599	80	973	344
(“offline networks” OR “online networks” OR “online communication networks”	14906	121	711	422	80	63	810	25	3820	96	29	15

OR offline communication networks”) AND (“software development” OR “information systems development”)												
TOTAL	23171	**353**	12394	**867**	1466	**114**	5469	**486**	12784	**229**	1457	**407**

2.4.4 Study selection

Following the high number of hits generated and respective reviewed papers, the reviewed papers were later filtered. First, papers whose title, abstract, introduction and conclusion fail to capture social network studies were excluded, and secondly, those papers that addressed social network structures and performance-related outcomes were included. In other words, relevant selected papers were included and discarded the irrelevant ones that deviate from the scope of the research purpose. The detailed criterion for inclusion and exclusion of journal studies is presented in the table below. Forward and backward searches for important articles was also conducted. This resulted in 2488 review papers, however, only 163 relevant journal articles were finally selected.

Table 5. Inclusion and exclusion criteria

Inclusion criteria	**Exclusion criteria**
Studies published between 1990-2020 SNA studies in knowledge-intensive and software development settings Studies that are focused on multiplex networks. Studies published in the English language. Scholarly (peer-reviewed) journals, including conference papers. Studies in Full text and PDFs	Studies not published in the English language. Non-scientific journals, e.g., keynote speeches, editorials, presentations, panel and editors' reports, experience reports, call for papers, research in progress, lesson learned. Duplicate articles and short papers (below 6 pages) Studies not available in full-text and PDFs Not scholarly (peer-reviewed) journals.

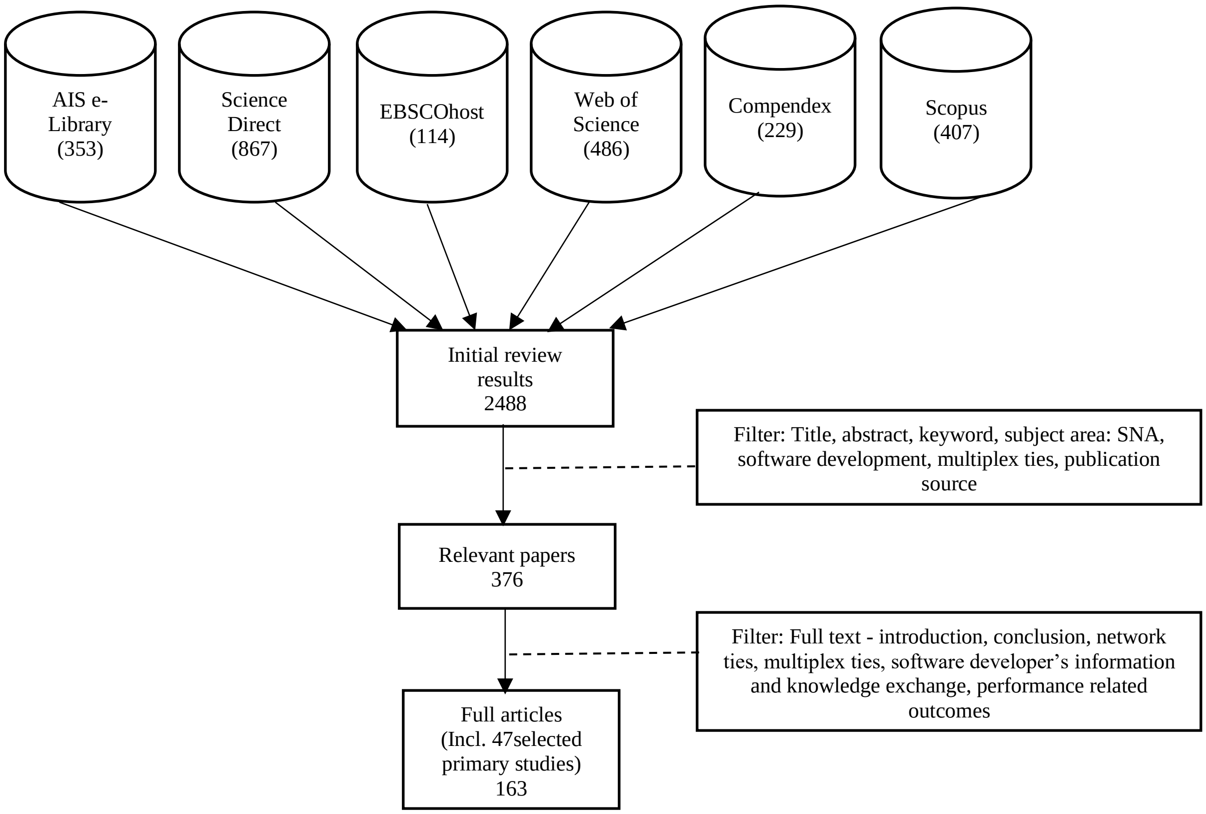


Figure 3. The systematic literature search of relevant articles

What follows is the review of the theoretical and empirical research on network multiplexity. Based on an integrative framework that depicts the relationships between one's social embeddedness in a network and its antecedents and consequences in the context of software development performance. The studies reviewed here were identified based on this systematic search of published research over the past 20 years (1990-2020) that have examined at least one of the linkages identified in figure 2.

The process resulted in three broad streams of research. Stream I pertain to research on the individual and organisational antecedents of network embeddedness. Stream II addresses the dynamic nature of networks relative to multiplexity. The dynamics of network evolution have been receiving increasing attention over the past decade, e.g., reviewing how networks change and adapt both at the individual (Aalbers, 2019; Buskens

& Van De Rijt, 2008) and organisational (Gulati & Gargiulo, 1999; Parker et al., 2016; Rowley et al., 2005; Shipilov et al., 2006) levels, yet relatively little is known about the mechanisms through which intraorganisational ties become multiplex. Stream II as such reviews the role of network dynamics – inserting a temporal dimension to our review of a multiplex tie in a software development context. Stream III addresses the variety of multiplexity angles covered. With a traditional view on multiplexity comprising of different resources provided between the same individual across their participation in different – yet parallel - networks e.g., simultaneously engaging on technical advice, friendship, day-to-day business and career help (Aalbers et al., 2014; Belso-Martínez et al., 2017; Lazega & Pattison, 1999; Snijders et al., 2013); others have interpreted multiplex networks to mean complementary use of offline and online communication networks (see for example Ding et al., 2019; Filiposka et al., 2017; Kane et al., 2014; Mesch & Talmud, 2006; Pavan, 2014; Zhang & Venkatesh, 2013; Zhang et al., 2008). Stream III hence reviews the role of media of interaction as a primer to multiplex interaction. The following paragraphs provide a review of the research that belongs to each of these streams.

2.5 Review of Literature

2.5.1 Multiplexity in Software Development

The basic idea of social network theory has focused on the various dimensions of social connections, brokerage roles, and resource benefits (e.g., access to knowledge and information) between people and organisations and what it means for network outcomes such as performance (Freeman, 1978; Wellman, 1983). A social network is a collection of relationships and ties that exist among individuals, teams, and organisations generally masked as social actors. Ties indicate the presence or absence of social connections in the relationship, for example being a friend of, or talking to specific persons, and are measured based on resources that flow through them (Aalbers et al., 2014; Bolander et al., 2015; Kane et al., 2014). Social ties provide learning opportunities and advantages to shape decisions and to access and utilise different resources that can affect project outcomes (Bolander et al., 2015). Research on social networks has focused on issues concerning how the environment shapes the way social actors perform tasks (i.e., "*Environmental shaping*"); the way movement of resources influence them (i.e., "*Contagion*"); the enabling benefits or constraining effects of social relationships to individuals (i.e., *"Social capital");* and how they derive benefits based on network access to resources (i.e., *"Resource access")* (Kane et al., 2014).

Social network analysis (SNA) is predicated on social network theory and provides a deeper visualisation of social networks. Ceci & Iubatti (2012, pp. 569–570) view SNA as techniques of mapping and measuring "*formal and informal relationships to understand what facilitates or impedes the knowledge flows that bind interacting units: i.e., who knows whom, and who shares what information and knowledge with whom, by what communication media.*" SNA examines social networks to inform practitioners and scholars about different social ties embedded in interpersonal relationships through which resources are enabled or constrained to achieve individuals and teams' performance (Ceci & Iubatti, 2012; Helms & van Reijsen, 2008; Nootjarat et al., 2015; Yang & Tang, 2004; Zhu & Watts, 2010).

A recent systematic literature review of SNA in software development projects (Schreiber & Zylka, 2020) reveals that deep multiplex examinations of software development environments are lacking. Yet, multiplexity has been shown in different contexts to produce beneficial results to individuals and their social environment (Burt, 1984; Coleman, 1988; Ibarra, 1995; Minor, 1983; Roberts & O'Reilly, 1979; Rogers & Kincaid, 1981; Smith-Doerr et al., 2004). This paper argues that a focus on the intertwining of formal and informal ties, across different types of knowledge resources, shared (e.g., business, technical or personal), and their mode of communication (e.g., offline, online) can reveal much about why variances in individual and team performance exist. Firstly, what multiplex means need to be defined in software development settings. For example, Snijders et al., (2013) describe uniplex networks as one-mode networks and multiplex networks as two-mode networks for their differentiation (see table 6). Yet, there exists a grey area unearthed for scholarly insights on what constitutes multiplexity and its potential benefits in social networks within multiplex debates.

In what follows the review of the related literature on software development and multiplex ties, applying a multiplex network perspective to understand performance in this context. Organised across 3 streams of literature by providing an overview of key concepts, identifying major relationships within and between these streams, and reviewing any conflicting evidence. The outcome of this review helps us to identify gaps in the research and avenue for future advancements in the IS domain in the topic.

IS-Based Network Research									
Author	**Journal/Conference**	**Data**	**Sample**	**Network Measures**	**Network Ties Embeddedness**	**Online or Offline**	**Network Outcome (DV related)**	**Major Findings**	**Network Orientation**
Yang and Tang, 2004	Information and Management	A longitudinal study: Teammate interaction questionnaire	25 teams in system analysis and design course	Centrality, Cohesion	Advice networks	ND	Team performance	Group cohesion was positively related to team performance.	One mode network
Ong et al., 2005	Proceedings of the 9th Asia Pacific Conference on Information Systems	A cross-sectional study: A survey	103 respondents in 18 ISD Undergraduate student teams		Task knowledge, Expertise-contribution misfit	ND	Team performance	Matching expertise to task knowledge contribution was more favourable to team performance. But negative when there was a misfit in expertise and level of contribution.	One mode network
Cataldo and Herbsleb, 2008	Proceedings of the ACM Conference on Computer Supported Cooperative Work	A longitudinal study: Interviews, Online chat, MR-tracking system	Group of software developers in geographically distributed projects	Degree centrality	Communication networks: Core persons as technical gatekeepers	Online	High productivity: as a development effort	Core developers were very technically competent and productive in the software	One mode network

								development effort.	
Helms and van Reijsen, 2008	Proceedings of the European Conference on Knowledge Management	A cross-sectional study: Electronic network survey	79 employees from an international product software company in the Netherlands	In-degree and Out-degree centralisation , Transitivity, Density	Knowledge networks, learning relations, knowledge transfer	ND	Group performance	There was a positive correlation between group performance and efficiency but a negative for transitivity. But negative correlation or marginal association between density and group performance - A more diverse but not redundant contacts.	One mode network
Xu and Ramesh, 2008	Journal of Management of Information Systems	A cross-sectional study: Survey experiment	132 participants in an experimental group at a university in the U.S.		Knowledge support: Generalised knowledge, Contextualised knowledge	ND	Process tailoring performance: as software tailoring task complexity	Contextualised knowledge was better for process tailoring performance.	One mode network
Zhang et al., 2008	Twenty-Ninth International Conference on Information Systems	A cross-sectional study: Paper roster and Email	52 students in the People's Republic of China	Closeness centrality	Online networks, Offline networks	Both	Students' course performance	Students' interdependent interactions in offline and online	Two-mode networks

		network survey						communication networks were more strongly associated with higher students' course performance, conditioned upon high centrality in both networks than their separation.	
Hossain and Zhu, 2009	Journal of High Technology Management Research	A longitudinal study: Email or message boards' forums	OSS dataset from SourceForge.com provided by the Department of Computer Science and Engineering, Notre, Dame University	Density, and Betweenness Centrality	Bug reporting, fixing and knowledge sharing	Online	Coordination performance: as defects management activities, e.g., quality and timeliness.	On average, an increase in centrality and betweenness was associated with an increase in the quality of code. But an increase in density decreased the quality of code measured.	One mode network
Sawyer et al., 2010	Informal Systems Journal	A cross-sectional and longitudinal study Field-based - ,	60 ISD teams at 22 sites of Fortune 500 organisations		Patterns of team level social interactions: Ambassador, Scout, Coordinator,	ND	ISD performance	No singular identified network patterns but all five influenced and maximised	One mode network

		Mail-based-, and Phone-based surveys, Visits			Sentry, Guard			ISD performance. Their variations also led to varying ISD performance.	
Singh and Tan, 2010	Journal of Management Information Systems	A cross-sectional study: Concurrent Versioning Systems (CVS) repository, SourceForge mailing lists.	186 projects with more than 10 registered developers		Communication patterns: Stable structures and efficiency, informative value, cost of link formation	ND	OSS team network formation	Not all stable structures in communication networks were efficient and not all unstable structures were inefficient based on conflicting individual vs group's interest or position to generate value.	One mode network
Kane and Borgatti, 2011	MIS Quarterly	A cross-sectional study: A network survey	468 employees in 32 workgroups	Eigenvector centrality	IS proficiency at group level	ND	Workgroup performance	The alignment of centrally proficient team users of IS was positively associated with workgroup performance.	One mode network
Sarker et al., 2011	Journal of Management Information Systems	A cross-sectional study: A network survey	111 Responses of globally distributed student teams from U.S. and Norway	Centrality	Communication, trust	Both	Individual Performance	Communication centrality and trust centrality were positively associated with	One mode network

			technology-based firms					individual performance among distributed teams. But the effect of communication was mediated by trust.	
Singh et al., 2011	MIS Quarterly	A longitudinal study: Panel data	2,378 projects at SourceForge	Cohesion	Network social capital: Internal contacts, External contacts	ND	OSS project success: as the rate of knowledge creation	More successful projects were favoured by greater internal cohesion and moderate external cohesion and technological diversity among project members. Including a positive effect of direct and indirect external contacts on successful projects.	One mode network
Maheshwari et al., 2012	Team Performance Management: An International	A cross-sectional study:	192 software development teams		Social capability, Technical capability	ND	Software development team performance	The alignment between social and technical	One mode network

	Journal	E-mail, Web-based survey						capabilities was associated with an improved performance of the software development team.	
Nan and Kumar, 2013	IEEE Transactions on Engineering Management	A cross-sectional study: Data extraction from code repositories in CVS archives	OSS dataset from SourceForge.net	Centralisation	Collaboration network, Structural interdependency	ND	Open-Source Software (OSS) Project performance	A high level of structural interdependency in large teams was more favourable to project performance, and a low level of structural interdependency in small teams was more favourable to project performance. However, the centrality of teams with high structural interdependence was positively linked to project	One mode network

								performance than its negative impact on the low level of structural interdependence.	
Sasidharan et al., 2012	Information Systems Research	A longitudinal study: A survey questionnaire	702 responses from a large southeastern university in the U.S.	Degree: In-degree and Betweenness centrality	Individual networks, Team networks	ND	Postimplementation success	Centralised structures were positively related to implementation success: In-degree and betweenness centrality were positively related to high task impact and information quality.	One mode network
Peng et al., 2013	Journal of Strategic Information Systems	A cross-sectional study: SourceForge data dump	OSS data from SourceForge, net		Co-membership ties: Leader-follower type, Follower-leader type, Follower-follower ties, Leader-leader ties, Later-stage ties, Same-stage ties, Early-stage ties	ND	Open-Source Software (OSS) development success	The network ties of leader-follower type and follower leader type were better conduits of OSS development success. Including	One mode network

								network ties to projects of later stages than early stages of development.	
Ryan and O'Connor, 2013	Information and Software Technology	A cross-sectional study: Online survey	181 respondents in 46 software SMEs		Tacit knowledge as expert knowledge, Transactive memory systems (TMS)	Offlin e	Successful team performance: as effective software teams	TMS and team tacit knowledge both determined whether some teams are low or high performing teams. Good quality social interactions shaped the acquisition and sharing of tacit knowledge.	
Zhang and Venkatesh, 2013	MIS Quarterly	A cross-sectional study: A network survey	104 respondents from a Fortune 500 telecommunicatio ns company in the U.S.		Direct ties, Indirect ties, Offline network ties, Online network ties	Both	Job performance	Employees' resourceful and complementary interaction in offline and online networks were positively related to job performance.	Two-mode networks
Ehrlich and Cataldo, 2014	Proceedings of the ACM Conference on Computer	A longitudinal study:	55 Jazz software development teams from two	In-degree and Out-degree centrality	Technical leaders' communication structure:	ND	Team performance	Hierarchical and small world patterns,	One mode network

	Supported Cooperative Work	Archival data, Interviews	companies in the automotive industry		hierarchical and small world patterns, information sharing			more sharing of information by technical leaders, and their centrality in the network were associated with the productivity and quality of a team.	
Park and Lee 2014	International Journal of Project Management	A cross-sectional study: A survey instrument	135 project teams in two large IT firms		Dependence, Trust	Both	Knowledge sharing, Team project performance	Knowledge sharing was strongly related to dependence and trust between project team members. Dependence and trust were fostered by frequency of communication, perceived level of similarity in the project's value and expertise.	Two-mode networks
Schmidt et al., 2014	35th International Conference on Information Systems	A cross-sectional study: Paper-based questionnaire	81 software development teams of a large global enterprise software		Shared mental models, Back up behaviours	ND	Team performance	The high degree of shared mental models resulted from pair	

								programming, code reviews and automated testing coupled with mutual feedback and work support on complex tasks (i.e., backup), which were linked to team performance.	
Sykes et al., 2014	MIS Quarterly	A cross-sectional study: Online Sociometric network surveys, Archival sources	87 knowledge workers	Eigenvector centrality	Advice networks: Workflow advice, Software advice (*each as singular connections*); Workflow and software get-advice, workflow and software give-advice	ND	Post-implementation employees job performance	The embeddedness in software and workflow gives and get advice networks were positively associated with employees' job performance.	One mode network
Lee et al., 2015	International Journal of Project Management	A cross-sectional study: A survey questionnaire	126 project teams		Business professional's expertise, Technical professional's expertise	ND	Higher level of team social capital, Project team performance	Shared vision and trust had a positive influence on project performance. Communicatio n effectiveness has a direct influence on	One mode network

								the shared vision between business and technical professionals. Business professionals were found to have encouraged the formation of social ties between business and technology professionals compared to the service providing technology professionals.	
Licorish and MacDonell, 2015	Information and Software Technology	: Mining data from IBM Rational Jazz repository	146 practitioners from an IBM Rational Jazz platform	Density In-degree Closeness	Openness to experience, Agreeableness, Extroversion	ND	Knowledge diffusion	Top managers played vital roles in diffusing knowledge around and have exhibited openness to experience in the communication networks.	One mode network

								Openness to experience was not found to be a major influence on global teams' success. Task structure with greater team connection and communication reduced the chance of losing tacit knowledge among global software teams.	
Çaglayan and Bener, 2016	Journal of Systems and Software	A longitudinal study: Codebase extraction datasets, Software, and IDE software development platform	500 kLOC part of the 20-year code base of the enterprise software product and Eclipse projects	Degree, Closeness, Betweenness and Pagerank centrality	Collaboration network	Online	Software quality: as defect proneness software	Committers with many collaborators were linked to greater defect proneness in software modules. Change of source code was also another risk factor in software modules.	One mode network
Latorre and	Journal of Systems	A cross-	39 organisations	Density	Collaboration	ND	Team formation	Previous work	One mode

Suárez, 2017	and Software	sectional study: Experiments' simulations, Questionnaire	of industry partners		network, Social skills			between members and their social skills were considered compatibility factors to a new project team formation.	network
Kudaravalli et al., 2017	MIS Quarterly	A cross-sectional data: Network survey	484 respondents of 71 teams from high technology firms in the U.S.	Centralisation	Interaction networks: Design collaboration, Technical collaboration	ND	Coordination success	The nature of expertise to perform tasks determined either to adopt a centralised and or decentralised work structure and the embedding nature of collaboration either design and or technical on decreasing team conflicts and increasing successful coordination of software development teams.	One mode network
Park and Lee,	Proceedings of the	A cross-	270 project teams		Dependence and	ND	Knowledge	Trust and	One mode

2017	Thirty-Eight International Conference on Information Systems	sectional study: Questionnaire			Trust based on: Expertise of team members, relationship exchange (benefit/investment), project complexity and communication frequency between business and technology professionals		sharing, Team performance in IS project	dependence among team members were positively associated with the increased level of knowledge sharing, and thus, project team performance. Project complexity influenced dependence; communication frequency influenced trust; Expertise influenced dependence and trust; Relationship benefit, not trust, influenced dependence; while relationship investment influenced trust, not	network

								dependency.	
Šmite et al., 2017	Information and Software Technology	A cross-sectional study: Survey data, Focus group, Interviews	61 members of software development teams from Ericsson and ABB software companies		Team's knowledge networks: Product knowledge, Process knowledge, Work-related knowledge	ND	Software development success	Social capital and networking were considered to 'solving complex, unfamiliar and' or interdependent tasks.	One mode network
Bernardi et al., 2018	Journal of Systems and Software	A longitudinal study: Data extraction from versioning systems, Issue tracker comments, Mailing lists messages, IRC chat log	976 bugs from Apache HTTP, 3,503 from GNU CCC, 12,972 from Mozilla Firefox, 294 from Xorg Xserver	Degree and Betweenness centrality	Committers (i.e., developers) communication	Online	Fix-inducing changes	Increased level of communication between fix-induce committers reduced the number of fixes. Major bugs were fixed by committers who did not induce them. A small proportion of bugs were fixed by committers who directly induced them. A few bugs were fixed by committers	One mode network

								who did not directly involve in previous fixes on the files.	
Datta, 2018	Empirical Software Engineering	A longitudinal study: Observation data of a product developed on the Jazz platform.	70 teams, 900 developers of major industrial product	Reachability, Connection, Association, Clustering	Intrateam connections (or interactions)	Both	Software quality: as the number of defects	Higher levels of connections between team members were related to more defects whereas higher clustering was associated with fewer defects. Average degree connections were related to more defects. The average clustering coefficient was related to a few defects.	One mode network
Purvis and Zagenczyk, 2018	European Conference of Information Systems	A cross-sectional data: A roster network survey	64 employees and project managers from an information technology organisation in Southeastern U.S.	Degree centrality	Team members' psychological climate: Trust ties, Advice ties, Friendship ties, Expertise ties (*each as singular connections*)	ND	ISD project performance	Friendship and trust ties were positively associated with team members' psychological climate but not expertise ties. However, trust	One mode networks

								showed positive mediation between expertise and psychological climate. But negative when trust mediated advice ties.	
Datta, 2019	Proceedings-Asia-Pacific Software Engineering Conference, APSEC	A longitudinal study: Mining data from Gnome suite or repositories	125 Projects, 2313 developers, 632096 comments 14-385113 defects	Centralisation: Degree and Eigenvector centrality	Developer communication network: Information exchange, developers' influence	Online	Software quality: as the number of defects, quality work products	The higher centralisation of information was related to more defects and vice versa. The higher centralisation of influence was related to a few defects. The mere facilitation of interaction between team members does not sufficiently guarantee an increase in team outcomes or quality work products.	One mode network

Non-IS-Based Network Research									
Lazega and Pattison, 1999	Social Networks	A cross-sectional study: Sociometric network data, Interviews	Respondents are lawyers and Associates in Corporate Law Firm	Transitivity, Reciprocity	Coworking ties, Advice ties, Friendship ties	ND	Collective participation, Status competition, Quality service	The interplay of various resources or their tie interdependence, namely: coworking, advice and friendship ties resulted in collective action and reduction in status competition between and among lawyers and associates to deliver quality service to clients.	Two-mode networks
Haythornthwaite, 2001	The Information Society	A cross-sectional study: Self-reports, Phone interviews	13 students from a classroom in a school	Density, Centrality	Collaborative work, Exchanging Information, Advice, Emotional support, Socialising.	Both	Completion of assignments, Work collaboration, Diversity of ideas, Strong and Weak ties balance	Multiplexity fostered associations between strong and weak ties to learn and generate diverse ideas needed to present and	Two-mode networks

								achieve project completion.	
Mesch and Talmud, 2006	The Information Society	A cross-sectional study: A network survey, Interviews	987 adolescent's household population in Israel		Offline relationships, Online relationships. Diversity of topics and activities,	Both	Quality of social relationships	The quality of relationships was influenced by their shared diversity of topics and activities over time across communication media.	Two-mode networks
Ferriani et al., 2012	Strategic Organisation	A longitudinal study: A network survey, Face-to-face interviews	80 firms from the Italian multimedia cluster in Italy	Degree centrality, Reciprocity, Transitivity, Balance	Economic ties, Social ties	ND	Multiplex formation	Social interaction and economic exchange were positively related to multiplex ties. Social ties moderated multiplex formation than economic ties.	Two-mode networks
Snijders et al., 2013	Social Networks	A longitudinal study: A three-wave network panel design,	75 MBA students	Degree centrality: Outdegree Indegree, Reciprocity,	Friendship and Advice-related interactions	ND	Shared employment preferences	There was a strong positive interdependenc e between friendship and advice relations	Two-mode networks

		A self-report, questionnaire		Transitivity.				on multiplexity but a negative effect on the specialisation at the actor level.	
Aalbers et al., 2014	British Journal of Management	A cross-sectional study: A semi-structured interview. A network survey	114 respondents from a European multinational electronics and engineering conglomerate. 281 respondents from European financial service provider in the Netherlands	Closeness centrality	Formal ties, Informal ties, Multiplex ties	ND	Innovative knowledge transfer	Closeness centrality in the overlap of formal and informal ties (i.e., multiplex ties) was positively associated with new innovative knowledge transfer.	Two-mode networks
Autry et al., 2014	Journal of Business Logistics	3 cross-sectional studies: A relational and process-based questionnaire	176 respondents from Retail companies in the U.S.	Triadic closure, Reciprocal connectivity, Homophily, Propinquity, Relationship tenure	Multiple relational, Process-based linkages	ND	Supply chain performance	Supply chain connectivity based on multiple sharing of relational – and process-based linkages was positively associated with improved chain performance. Processed-based linkages was a major	Two-mode networks

								mediator variable.	
Lee and Lee 2015	Management Communication Quarterly	A cross-sectional study: Sociometric network data	53 respondents from an Advertising agency in Korea	Reciprocity, Homophily	Advice ties, Knowledge ties and Friendship ties	ND	Creative interaction	Creative interaction ties consisted of diverse dimensions of intra-organisational relations: friendship ties on access to opportunities and efficient transfer of information,	Two-mode networks
Belso-Martinez et al., 2017	Entrepreneurship & Regional Development	A cross-sectional study: Interviews. Network questionnaire	28 business firms in the Biotech cluster of Alicante in Spain	Structural hole	Technical relations, Business relations, Multiplex relations	ND	Innovation performance	Access to diverse knowledge via structural holes in technical and business networks was positively linked to innovation performance.	Two-mode networks
Methot et al., 2016	Personnel Psychology	A longitudinal study: Network survey	168 employees from Insurance company 182 employees from Restaurant and retail sales		Affective and Business relationship, Workplace friendships	ND	Job performance	Multiplex friendships were positively associated with employee job performance	Two-mode networks

								but were partly challenged by emotional exhaustion but then offset by the increased level of positive affect present.	
Hood et al., 2017	Journal of Business and Psychology	A longitudinal study: A social network business simulation	538 undergraduate business students at a public university in the U.S.		Task conflicts with friends, TCs with Non-friends, Multiplex friend-TC; Relationship conflicts with friends, RCs with Non-friends, Multiplex friend-RC	Both	Team performance	Relationship conflicts among friends in a team were positively linked to team performance, whereas those occurring among non-friends in a team was negatively associated with team performance.	Two-mode networks
Cai et al., 2018	Physica A: A Statistical Mechanics and Its Applications	A sampling network survey	234 respondents from 3 small and medium-sized firms in Xi'an hi-tech industrial development zone in China	Degree centrality, Eigenvector	Formal and Informal relations, Multiplex relations	ND	Job performance	Degree centrality and eigenvector in the multiplex network were more accurate predictors than each network	Two-mode networks

								layer, of employee performance.	
Marineau et al., 2018	Journal of Business and Psychology	A cross-sectional study: Sociometric questionnaires	75 employees from a medium-sized Midwestern US life science		Task conflict, Relationship conflict	ND	Advice Seeking	Employees were more able to seek advice with whom they have task conflict despite the presence of relationship conflict. But unlikely to seek advice when relationship conflict is absent.	Two-mode networks
Ding et al., 2019	Fortieth International Conference on Information Systems, Munich	A longitudinal study	365 students from an Asian university	Reciprocity	Online communication, Offline communication	Both	Student academic performance (i.e., GPA)	The overlap of online and offline connections showed significant correlations with higher students' GPAs. Online communication between two students has a positive influence on online	Two-mode networks

								communication but negative impacts otherwise. Less reciprocity across the two media but more reciprocity over a single medium.	

Table 6. Selected Primary Studies of SNA and Conceptualisation of Resource Relations in IS and Non-IS Research

2.5.2 STREAM I: Individual and organisational antecedents of network embeddedness in Software Development

Irrespective of how software developers relate to one another and the impact of such relationships on information and knowledge resources sharing, organisations, in general, find it difficult to drive successful software development (Koivuaho & Laihonen, 2006; Sasidharan et al., 2012; Sawyer et al., 2010). Social interactions in software development related to how employees work, communicate, coordinate and manage conflicts within and between a team and non-team members for information and knowledge resources to develop software products (Long & Siau, 2008; Sawyer et al., 2010). Software development comprises activities about analysing user requirements, designing and writing relevant codes, developing, verifying and testing features before releasing them into production as final software products, and its further maintenance for clients (Singh & Tan, 2010; Tang, 2015; Ulhas et al., 2016). The series of software development tasks makes this process a complex and knowledge-intensive activity, whereby diverse specialised knowledge, skills, and the experience of team members is brought to bear to achieve successful software projects. The complex nature of this process requires team members to be creative, communicative, informative and knowledgeable to tackle variations and unpredictability that inheres in software development (Barthelmess, 2003; Cataldo & Herbsleb, 2008; Lindsjørn et al., 2016; Sulayman et al., 2012; Tang, 2015). Indeed, research posits that software development teams require different knowledge skillsets ranging from domain knowledge, technical knowledge (e.g., for programming/methodologies), process knowledge and product knowledge (e.g., coding source/architecture), in order to manage software requirement tasks (Maheshwari et al., 2012; Ryan & O'Connor, 2013; Šmite et al., 2017). Overall, different knowledge interactions are required to manage the business, customer and technology side of the software development process, leading to improved productivity and quality of software projects (Xu & Ramesh, 2008).

2.5.2.1 Individual antecedents

Given that people are important to managing the input and output demand of software development tasks, team members are challenged to collaborate by way of information and knowledge sharing to manage this work complexity and to deliver shared project performance goals (Barthelmess, 2003; Jordan & Segelod, 2006; Lindsjørn et al., 2016; Purvis & Zagenczyk, 2018; Ulhas et al., 2016; Xu & Ramesh, 2008). This has led to researchers considering the human factor as playing a huge role in the software development process (Nidumolu, 1995; Zhang et al., 2019) and overall software quality (Datta, 2018). In light of the reliance of software development work on myriad and interdependent knowledge and skills of individuals (Kudaravalli et al., 2017; Long & Siau, 2008; Ryan & O'Connor, 2013), galvanising an effective knowledge network is a key enabler for software developers to succeed in software projects (Helms & van Reijsen, 2008; Hossain & Zhu, 2009; Šmite et al., 2017). For instance, and particularly relevant in a software development setting where typically, developers work together to develop the breakdown of work (e.g., scrum), extra knowledge available allows one to better determine and interpret how others will behave in one context if their behaviour and attitude is known from a different context. In other words: a relation of one kind keeps in check the negative side-effects of a relation of a different kind (Albrecht & Ropp, 1984; Marineau et al., 2018; Marsden, 1981). Studies in different settings have found that ties that combine multiple dimensions of a relation between two concrete individuals can have a substantial and qualitatively different effect in comparison to the effects of their constituting elements (Burt, 1984; Smith-Doerr et al., 2004). However, little attention exists as to how the diverse skills and abilities of individual software developers affect how their teams interact and coordinate software development tasks (Šmite et al., 2017). Hence, a multiplex network structural perspective can better address this gap.

2.5.2.2 Organisational antecedents

For organisations to achieve software development success, a cross-functional effort is critical to software developers' work effectiveness (Licorish & MacDonell, 2017; Rondeau et al., 2006). However, issues surrounding software development performance are strongly associated with organisational communication, knowledge, control, work efficiency (i.e., time and budget costs to develop and deliver software products) and effectiveness (i.e., the fit of software applications and adaptations to user requirements and achievement of overall project goals) among software developers and other stakeholders. Hence work efficiency and effectiveness are indicative measures of software team performance (Jiang et al., 2004; Ryan & O'Connor, 2013). Interestingly, software developers' performance is also characterised by an individual's quest to network, collaborate and achieve success (Licorish & MacDonell, 2017; Peng et al., 2013; Rasch & Tosi, 1992; Singh et al., 2011). For example, scholars find that the personality (e.g., openness to experience) of individuals influence their knowledge diffusion among members (Licorish & MacDonell, 2014). Furthermore, extraversion, another personality trait is also linked to high interaction among agile developers and eventually driving software quality (Acuña et al., 2009). This may be salient to why software developers communicate, collaborate and coordinate work with others to access pertinent knowledge resources and in turn increase work performance (Datta, 2018; Licorish & MacDonell, 2017; Mishra et al., 2012; Šmite et al., 2017).

2.5.2.3 Multiplexity dimensions

Over two decades of multiplex network studies have shown the different dimensions of multiplexity from purely resource focused to include communication network and relationship perspectives. Each of these without their integral view, however, have limitations to getting a deeper understanding of the dynamics of network structures and their differential outcomes (e.g., Liu, Park, Hymer, & Thatcher, 2019; Methot & Rosado-solomon, 2019). Further details are below.

Resources Perspective

This perspective focuses on the diversity of resources that flow through social networks. It emphasises that resources are not limited to the aggregation of information and knowledge but their types. Cross et al., (2001, p. 231) suggest that *"A disregard for additional contents, together with the demands of convenience, often dictates that we collect data on only the broadest of relations"*. The precise nature of what flows through social networks is often unclear in network research studies. While this perspective focuses on diverse resource types and uses, such exchanges are limited to single relationships and fail to account for multiple exchanges within the whole network. Hence the loss of valuable information data types (Bliemel et al., 2014; Meng et al., 2016), and currently limiting sufficient understanding of resource relations in software developer networks (Schreiber & Zylka, 2020).

Usually, employees relate with diverse resources (work or non-work related) by maintaining ties to different network members. For example, the use of business knowledge or workflow advice and technical knowledge or software advice is important information and knowledge resources needed among software developers (Belso-Martínez et al., 2017; Sykes et al., 2014). The integration of business and technical competencies have been acknowledged as key to leading innovative solutions (Whelan et al., 2011). In essence, social (i.e., affective/expressive) and economic (i.e., instrumental) resources can be shared to meet different needs (Ferriani et al., 2012; Gimeno & Woo, 1996). Scholars view that embedding social networks where software developers operate often determine the extent and diversity of resources accessible to them (Schreiber & Zylka, 2020) to achieve high performance (Peng et al., 2013). Differentiating business, technical and personal related resources upon which individuals are embedded can help resolve vagueness in resource relations between software developers.

Communication Networks Perspective

This perspective relates to the network media through which work and non-work-related resources get created, accessed, and shared to get work done. This has two modes: offline (i.e., face-to-face) and online (i.e., web/internet) means of communication among individuals and teams. With the advancement in information technologies (IT), this perspective focuses on technology use and how it influences people's ability to achieve work collaboration, efficient communication, and performance. This makes offline and online interactions between people important to studying social networks (Filiposka et al., 2017; Haythornthwaite, 2001; Mesch & Talmud, 2006; Pavan, 2014). Extant studies reveal that these networks if rightly harnessed can facilitate communication and coordination of tasks and software quality (Bernardi et al., 2018; Hossain & Zhu, 2009; Storey et al., 2017; WeiQiang et al., 2014). However, social theory scholars argue that the nature of resources being shared and work structures of individuals in these media tend to be blurred to understand the context by which media capability is most effective (Ngwenyama & Lee, 1997).

Relationships Perspective

This perspective describes the natural connections of people across formal and informal relationships that may occur in a workplace environment. This may lead in the direction of cooperative and/or competitive orientation, and across vertical and horizontal connections among employees (Aalbers et al., 2014; Gimeno & Woo, 1996; Maggioni et al., 2013; Shipilov, 2012; Shipilov et al., 2014). A multiplex relationship perspective is also structured exclusively around members' positions and connections that tend to only channels aggregate or combine resources in the network. However, this perspective also fails to capture diverse resource interactions among many social actors (Bliemel et al., 2014).

In review, work on the individual and organisational antecedents of network embeddedness in software development has revealed the important roles human interactions and their social characteristics play in organisational knowledge sharing and application when developing software products, which reviewed studies show are key to managing the already complex nature of software development process. These studies emphasised that individual knowledge roles and their concerted efforts in teamwork activities shapes overall project performance. However, how the individual knowledge, skills and attitudes of software developers help them to communicate and mobilise their knowledge networks to secure diverse resources and to perform and coordinate tasks to the team's advantage is underexplored. This review seeks to advance research in that direction based on the dimensions of multiplexity to better understand network embeddedness in software development.

2.5.3 STREAM II: Software Development as a Social Dynamic Process

Recent studies suggest that the continuous replacement of single plan based traditional software development method by agile method has charged software developers to be flexible towards managing emerging developments (such as changes in requirements) if a quality-driven process is to be maintained (Schmidt et al., 2014). As phases of software development generally revolve around requirements analysis, design, implementation, testing and deployment (Koivuaho & Laihonen, 2006; Ong et al., 2005; Schreiber & Zylka, 2020; Tang, 2015; Ulhas et al., 2016), the network ties by which software developers manage tasks dependencies are important. For example, Peng et al., (2013) find that software developers would need to form and utilise different network ties, such as leader-follower and follower-leader ties, to successfully maintain work effectiveness throughout software development stages to achieve project success. While each of these phases tends to be unique and may require different expertise and knowledge resources, these are mostly interdependent which often result in the unintended consequence of silos developing among developers and teams (Mehta & Bharadwaj, 2015; Ong et al., 2005; Peng et al., 2013). Sharing a diversity of knowledge, such as technical and business-related during social relationships is needed for effective integration. Depending on the nature of tasks and development stages, the shared knowledge relations may need to change as the nature of tasks and projects changes with requirements (Cataldo & Herbsleb, 2008; Licorish & MacDonell, 2014; Sawyer et al., 2010; WeiQiang et al., 2014; H. L. Yang & Tang, 2004). In this case, it would be vital to know at what development stage that two or more knowledge relations would be quintessential, for example from maintaining purely technical knowledge ties to adding business knowledge ties, etc. – a transitioning from a uniplex to multiplex tie between two developers. Understanding social ties dynamism among software developers is of interest to make sense of multiplexity in software development environments.

Scholars posit that situational factors (e.g., scarcity of resources, project reorganisation) can cause relationships to experience changes over time that may force individuals to form new ties with other people, replace the old ones, or maintain existing ties depending on where specific resources are available and accessible (Bliemel et al., 2014; Hite, 2005; Koka et al., 2006; Venkatraman & Lee, 2004). Given that access to relevant resources can influence network formation, the relationships upon which these resources flow have greater embeddedness in multiplex ties than uniplex ones (Lee & Monge, 2011). As the scope of a task increases with complexity, how two software developers maintain multiplex ties would have implications on how they perform on the job. Schreiber and Zylka (2020) claim that the kind of networks that software developers maintain to access diverse resources will dictate their level of productivity. In other words, different network ties come with different resource benefits (Belso-Martínez et al., 2017; Singh & Tan, 2010), and so changes within and across them may impact individuals' network outcomes (Bliemel et al., 2014). For example, extant studies show that not all software developers' communication structures can maintain stability and efficiency on completing tasks because some of them can be unstable yet be efficient while the stable ones may not necessarily be efficient throughout the development stages due to shifting personal interest and differing social status (Singh & Tan, 2010; Xuan et al., 2012). Related research has that stable communication structures are more suited for high execution and repetition of tasks, whereas unstable ones are more suited for creative work (Gloor et al., 2007). This suggests that software developers would be expected to dissolve, maintain or form new ties to secure performance-enhancing resources. For example, studies reveal how receiving positive performance feedback can make individuals change their network ties to sustain performance (Parker et al., 2016). Therefore the flow of information advice is conceived to be a cause of network formation and change (Ferriani et al., 2012).

Likewise, studies link identity and calculative orientation of two people towards tie formation (Hite & Hesterly, 2001). This may not be generalised to multiplex ties among actors, given that multiplex ties can also be motivated by structural commonalities (relational or operational positions), assortative commonalities (compatibilities in actors' attributes) and proximity-based commonalities (a link between interfirm relational–or process-based interaction) based on location and co-location over time (Autry et al., 2014). As has been said, social skills have been considered as attributes to software team network affiliation, a uniplex tie (Latorre & Suárez, 2017). Similar studies link similarity and shared third party members to forming multiplex ties (Autry et al., 2014; Mesch & Talmud, 2006; Newig et al., 2010; Shipilov, 2012; Shipilov et al., 2014). Since social and economic motives are central to all forms of emerging relationships (Autry et al., 2014; Ferriani et al., 2012; Hite & Hesterly, 2001), Ferriani et al., (2012) claim that not all uniplex ties will be elevated into multiplex ties because some actors will maintain purely economic or social ties and not both, arguing further that any of these ties can dissipate over time. In software settings, the extent to which software developers maintain economically or socially related resources or both can inform our understanding of the conditions and the temporality that may warrant how multiplex ties may develop and or dissolve through time. Whether multiplexity should be considered as an asset rather than a liability given that the nature of software task is dynamic and can influence network affiliations. Despite multiplex ties appearing as integrative network structures and tends to have a greater impact on task performance than purely uniplex ties, the aforesaid changes in software development stages and projects might weaken its strength as shared interests and tasks orientation also widens, and the social familiarity and relational maintenance diminish between two individuals (cf. Gloor et al., 2007; Singh & Tan, 2010; Xuan et al., 2012).

Furthermore, Lee & Monge (2011) signal an inconsistency in the network relationships on how knowledge is shared and implemented in community-based projects. This may be partly connected to how ideas or knowledge sharing between actors are managed to realise for example innovations in project development. Because the lack of social ties between the person who source ideas and the one who gets them to be implemented can pose a barrier to innovation development (Whelan et al., 2011). This is of interest to examine multiplex formation within software development where social ties' embeddedness matter for software designs and implementation in the project development process (cf. Granovetter, 1973; Ferriani et al., 2012). Shipilov et al., (2014) argue that a strategic perspective is warranted to interpret ties dynamics in organisations and how people are embedded in various relationships, their interdependence, and how they shape organisational activities. In sum, different resources being shared, shared partners and network positions are generally linked to network formation including multiplex ties (Lee & Monge, 2011). Considering the complex nature of multiplexity and its dynamic operational processes, including the complexity of tasks in software development – software developers' attributes and endogenous relationship patterns in social and knowledge relations can inform research on multiplex tie formation and dissolution in the software development process (cf. Singh & Tan, 2010). Overall, Rosenkopf & Padula (2008) posit that network changes can occur radically (emergent of new network structures) or incrementally (network reformations or reorganisations) that may affect individuals' social ties and differentials in project work outcomes.

In review, work on software development as a social dynamic process revealed that the nature of software development tasks demand software developers interacts with similar and different knowledge workers with business and technical expertise that each stage of the development process deserves. Generally, studies revealed that individuals sharing similar social attributes, location, structural position, and direction of informational resources (e.g., advice) influence the emergence of uniplex and multiplex ties. However, this review overall showed a lack of understanding under what conditions the social and economic or instrumental orientation that people share in their work roles and knowledge relations with others leads to multiplex ties. Plus, the temporality of such ties is due to changes in software development stages and projects over time. This review considers this avenue worthy of further research in software development where the nature of tasks is mostly dynamic across different development stages and may depend on separate or parallel knowledge interactions and integrations to get work done among software developers.

2.5.4 STREAM III: Channel Multiplexity in Software Development Research

As aforesaid, multiplex ties partly constitute an overlapping of formal and informal relationships. The former relates to business, economic or instrumental exchanges between coworkers, while the latter relates to affective or expressive exchanges between friends. Their coexistence in an individual's network provides access to diverse information and knowledge resources, in terms of work (material) and non-work (morale) related benefits. Since resources may vary across different contexts (Lazega and Pattison, 1999; Lee and Lee, 2015; Methot *et al.*, 2016; Marineau et al., 2018), the case of multiplex ties means that formal and informal relationships complement each other for initiating and performing work (Aalbers et al., 2014). Recent studies conceptualise formal and informal structures as "*communicational ambidexterity*", which relates to different communication needs among individuals. This is typified as *"univocality"* (i.e. formal - top-down approach, centralised mode of communication) and "*multivocality*" (i.e. informal-bottom up approach, participative or decentralised mode of communication) (Huang et al., 2015). The communication networks are often centrally and decentrally structured when considering formal and informal communication needs in software development (Long & Siau, 2008). This informal communication structure is deemed to be self-organised and coevolved, which is why communication patterns are important for sharing information and knowledge resources among software developers (Singh & Tan, 2010). However, how individuals navigate through informal structures often determine how they can marshal information and knowledge resources to realise their goals (Marineau et al., 2018) and product development success (Ehrlich & Cataldo, 2014).

The position of multiplex ties, whether they emerge coincidentally or are purposefully orchestrated, is mixed in extant studies (Fujimoto & Valente, 2015). For example, Kim & Goh (2013) claim that they coevolve for collective action. Other studies show how networks are generally orchestrated by emergent phenomena – a combination of the two views (see for example Clegg *et al.*, 2016). A multiplex network may therefore consist of heterogeneous relationships with diverse expertise yet belong to similar or other groups (Hood et al., 2017). Where a multiplex tie exists, members tend to enjoy diverse resources, greater communication, commitment and trust, quality feedback, positive energy, emotional and work support (Heaney, 2014; Hood et al., 2017; Lorenzen & Vaarst Andersen, 2012; Methot et al., 2016); reciprocal exchanges, shared value, converging interests, collective solidarity, resource stability (Rogan, 2014; Tuli et al., 2010); robust structures to limit structural constraints, and resilience to relationship breakdown in case of conflicts (Marineau et al., 2018; Min et al., 2014; Subramaniam et al., 2013). These benefits can certainly increase performance among software developers.

A few scholars contend that multiplex ties may result in informational redundancies but the roles convergence seems to enable actors to pursue and bridge for resources across homophiles and structural holes where strong ties can be weak ties and vice versa (Clegg et al., 2016; Heaney, 2014; Verbrugge, 1979). That is, with strong ties, individuals can gain closure on existing resources and opportunities particularly in stable conditions. While with weak ties, individuals can access new resources, however difficult when access to diverse knowledge may be scarce (Belso-Martínez et al., 2017; Granovetter, 1973; Shipilov, 2012). Homophiles are associated with social similarity – in terms of background, age, gender, cultural norms and values – that makes it easier for members to trust, communicate and coordinate activities together (Lee & Lee, 2015; Mesch & Talmud, 2006).

Adhering to Resource Dependency Theory (RDT), studies report that diverse information and knowledge resources are required for various tasks to drive group performance (Ashworth & Carley, 2006). RDT advocates for strategic access to stable and diverse resources (Berrett & Holliday, 2018). For example, Belso-Martínez et al.,

(2017) examine interdependences of business and technical networks and find that members' brokerage across structural holes in both networks channelled access to business and technical knowledge which eventually contributed to their firm's innovation performance. A structural hole is a network measure that relates to an individual's connection to two or more unconnected and non-redundant others (Obstfeld et al., 2014). In doing so, individuals can access and control quality ideas, new information and knowledge to inspire creativity and consequently innovation development. People with higher network contacts are said to have a greater number of structural holes to gain resource advantage (Kuegler et al., 2015; Belso-Martínez et al., 2017).

Lazega & Pattison (1999) investigate resource networks among partners and associates, whose nature of work is driven by cooperation and status competition. Their findings revealed that an attempt to drive quality service offerings to client influence each member's reliance on different social ties such as coworking, advice and friendship ties. Friendship ties are regarded stronger on local network clustering among friends and in advice exchange among seniors, a more hierarchical linkage. This research concluded that these resources together reduce the degree of status competition but foster cooperation among the partners and associates for collective action. The RDT perspective also suggests that rather than concentrating on limited resources, diversifying into different resources are important for individuals to reduce resource scarcity, instability and dependency on a few actors to truly foster greater performance (Berrett & Holliday, 2018). Active resource interactions and diversifications have been linked to promoting problem-solving and interpersonal learning to accomplish work, job satisfaction and organisational performance (Agneessens & Wittek, 2012). Multiplex relationships are therefore interdependent, reinforce each other, and provide a better understanding of local network structures (Lazega and Pattison, 1999; Lee and Lee, 2015).

2.5.2.1 Multiplex Media Embeddedness

Communication barriers are obstacles to successful software development among co-located and distributed software developers. Communication is needed to correct and prevent bugs or defects, retrieve software documentation, resolve design problems, and ultimately drive project goals. Hence scholars have recognised that software developers rely on ICT (e.g., email, internet relay chat, social media, etc.) for communication and coordination of information and knowledge resources beyond the face-to-face meetings, to drive performance (Bernardi et al., 2018; Cataldo & Herbsleb, 2008; Defranco & Laplante, 2017; Ehrlich & Cataldo, 2014; Park & Lee, 2017; Storey et al., 2017; Subramaniam et al., 2013; WeiQiang et al., 2014). Multiplex ties also partly embrace offline and online communication networking among individuals to create, mediate and manage resources and work relationships (Filiposka et al., 2017; Haythornthwaite, 2001; Zhang & Venkatesh, 2013). Numerous scholars regard different communication theories (e.g., media richness theory, media synchronicity theory, theory of complementarity, critical social theory (Dennis et al., 2008; Fox et al., 2010; Ngwenyama & Lee, 1997; Zhang & Venkatesh, 2013; Zhang et al., 2008) to understand the benefits of accessing and processing information and knowledge resources in offline and online interactions. Media richness theory takes the view that each communication medium offers different benefits based on parallelism, reprocessibility and rehearsability. Parallelism relates to the capability to share multiple information and knowledge resources simultaneously. Reprocessibility relates to the capability to re-evaluate shared information and knowledge resources, while rehearsability relates to the capability to revisit, re-edit and refine shared information and knowledge resources (Fox et al., 2010; Zhang et al., 2008).

For offline (face-to-face) communication, scholars posit that this medium offers richness as individuals can get personal attention, exchange verbal gestures, which also permits immediate response to sharing information and knowledge resources, especially tacit knowledge. As offline communication may benefit from parallelism, scholars posit that online communication (e.g., discussion platforms, emails) may be more suitable to document, track, re-evaluate, re-edit and share knowledge of explicit nature (Zhang & Venkatesh, 2013; Zhang et al., 2008). While online communication is generally effective for software developers to connect and commit to sharing work files, it helps distributed team members to resolve temporal constraints on project development (Park & Lee, 2014; Xuan et al., 2012; Zhang & Venkatesh, 2013). Scholars apply both the media synchronicity theory and theory of complementarity to support the idea of using offline and online (e.g., written documents, telephone) communication to meet different needs (Dennis et al., 2008; Fox et al., 2010; Zhang et al., 2008). Ever since the advent of the Internet, online communication via emails, Skype, social media has been integrated into social interactions (Contractor et al., 2011; Kane et al., 2014; Pavan, 2014; Wellman et al., 1996; Zhang & Venkatesh, 2013). In online communication, for example, social media is deemed an alternative means to offline communication and collective completion of work related to lead enterprise software users (Subramaniam et al., 2013).

Extant research further shows that social ties in offline and online communication networks involve direct and indirect ties, both of which have implications for the movement of information and knowledge resources among individuals (Zhang & Venkatesh, 2013). A case in point is that employees with direct (i.e., strong) ties in the offline relationship tend to quickly access relevant, contextual, and trusted information, and to communicate more. Such that it increases social familiarity but minimises misunderstandings, and so able to celebrate verbal signals. Considering similar characteristics of indirect (strong) ties in an online relationship, the issue of familiarity is lean and body language uncertain but promotes a high level of information transmission, documentation, and retrieval. It is asserted that employees with indirect (i.e., weak) ties in the offline relationship tend to experience less communication and maintenance costs in time but can gain broader network coverage. Similar patterns also occur among employees with indirect (weak) ties in an online relationship (Zhang & Venkatesh, 2013). While the strength of strong and weak ties might be dependent on the direct and indirect personal contacts across the communication networks, it is found that strong ties are more associated with offline communication and weak ties more dominant in online interactions. But that strong ties favours reciprocal exchanges over weak ties, and are more preserved across multiple layers (Filiposka et al., 2017). Studies reveal a positive link between software developers' direct and indirect internal and external network ties and software project success (Singh et al., 2011). Overall, scholars suggest that the quality of social ties should be more focused on social similarity, interactive intensity and resources shared between communicating individuals. Thus, multiplex communication networks are more focused on how technology is utilised to support physical meetings and share relevant resources that best deliver performance, not employees' shared characteristics (Mesch & Talmud, 2006).

Despite scholars conceding that overlap usage of offline and online networks affect performance differently (Ding et al., 2019; Zhang & Venkatesh, 2013), preliminary studies reveal that individuals may not actively reciprocate across these two networks (see for example Ding *et al.*, 2019). While this research was conducted among students, this conflicting view deserves further research, specifically when considering the direction of information and knowledge flow across the two media in software development. Pavan (2014) argues that rather than treating technologies as exogenous material outside social networks, its "*sociomateriality*" (i.e., social features) deserves a place in social networks research. Scholars assert that relationships affect the extent of IT use (not its materiality) and performance (Sarker et al., 2011). Since technology use is one of the key dimensions of success in software development performance, this study argues that multiplex ties may shape how such is harnessed or utilised among individuals (Kuegler et al., 2015).

In review, work on channel multiplexity in software development research revealed the different angles to viewing and applying multiplex view to the dynamics of software development. This considered the reality of formal and informal work structures that software developers operate across offline and online communication networks to discuss task-related needs, and access to diverse resources on performance. However, these studies which considered that individuals must maintain network ties in both networks to increase performance fail to consider the direction of diverse flow of resources. This includes the conflicting and insufficient understanding of the nature of reciprocity across these media, and how this can equally affect performance differentials. This also presents a research opportunity to shed light on these grey areas.

Above all, the metaphysical dimensions to which ties are created, linked, and changed across communication networks, relationships and resources can sway employees' performance during project development. While extant studies observe the relevance of each dimension, each element does not stand alone; they are all intertwined as forms of multiplex ties. Hence an outcome of this view of the multiplexity based view (herewith refers as MBV) revolves around the overlapping of various types of interactions across communication networks (e.g., online/offline), relationships (e.g., formal/informal) and resources (e.g., business, technical, personal support, etc.) between two actors towards meeting their specific needs (e.g., social, economic, or instrumental related), as graphically represented in figure 4 below. It should be noted that the nature of interactions can be collaborative and or competitive (Aalbers et al., 2014; Ferriani et al., 2012; Gimeno & Woo, 1996; Haythornthwaite, 2001; Lazega & Pattison, 1999; Mesch & Talmud, 2006; Shipilov et al., 2014). Given the latter, two people or organisations can achieve greater resource utilisation through *coopetitive relationships* – e.g., Oneworld airlines can serve airline customers more flexibly and productively through collaboration while there is also competition between the airlines within the network. Another interesting example is employees jointly working on projects to deliver network and organisational performance also vying for individual promotions, yet maintaining coopetitive or ambivalent orientation in their workplace (Rothman, Pratt, Rees, & Vogus, 2017). This multiplex-based view of social networks can increase understanding of the actual layering in software developer networks.

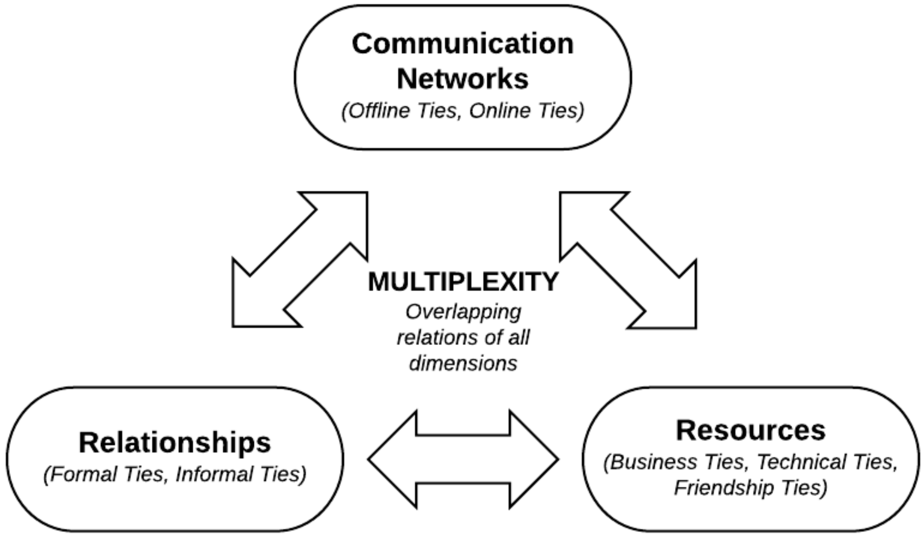


Figure 4. Multiplex Based View of Social Networks

The multiplexity graph shows the reality of various layering of specific diverse resources being communicated in different individuals' relationships. Thus, providing a clearer perspective by which researchers can evaluate the actual communication profile of software developers and performance differentials. Table 7 below provide further considerations regarding the network dynamics of social networks. While this may seem complex and challenging for analysis (Meng et al., 2016), it still opens an opportunity for IS researchers to firmly understand software developer networks and performance differentials. Figure 5 below provides a representation of our literature review: Stream I - III, hence highlighting their interrelated relationships of the social dynamics of software developer networks from a multiplex network perspective.

Table 7. Variations of Network Embedding Characteristics in Multiplex Network Ties

Multiplex Network Ties	**Variations of Network Embedding Characteristics**
Relationships	Software developers may involve in formal, informal, or multiple relationships (*overlap of formal/informal ties*). Different software tasks/roles may invite different network affiliations/memberships over time. Different network affiliations and activities or software tasks may attract diverse resources. Different network affiliations may underscore different activities or software tasks and can also dictate the control of diverse resources.
Communication Networks	Software developers may engage in offline, online, or multiple media interactions (*overlap of offline/online ties*) to access diverse resources. Different network affiliations and activities or software tasks and sources of resources may influence media choice/use. Different communication media can also shape software developer activities/ affiliations.
Resources	Software developers may interact with business, technical, friendship, personal or multiple informational and knowledge resources (*overlap of two or more resources*) to perform various roles, activities, or software tasks. The different flow of resources may shape or alter network affiliations/memberships over time. The flow of resources can foster or forestall activities or software tasks. Work activities or software tasks can also affect the flow of resources and network affiliations.

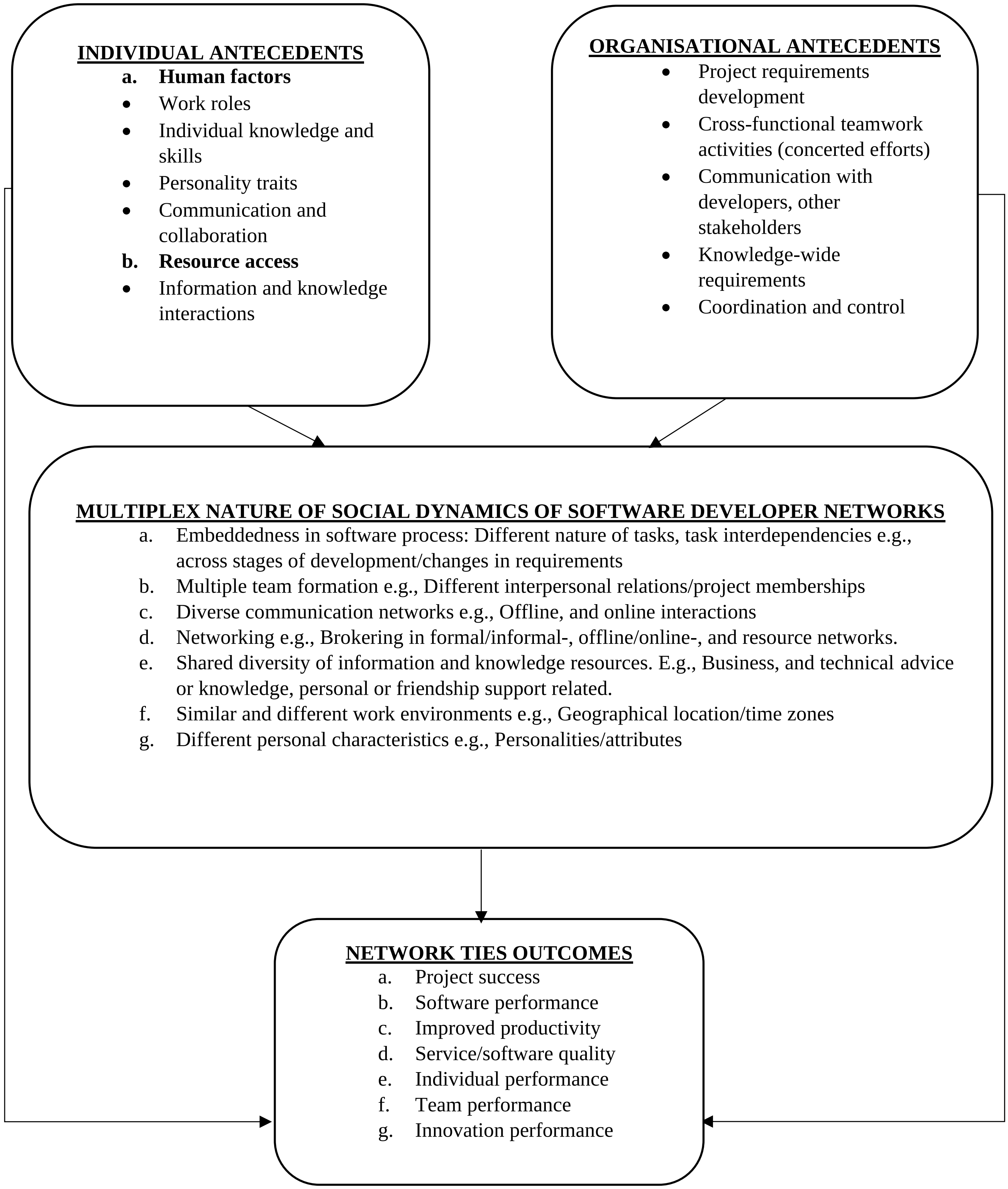


Figure 5. Interrelationships of The Research Streams on Social Dynamics in Software Development

Figure 5 suggests that for stream I, individual and organisational antecedents are interwoven in the sense that the quest for developing a software project requires understanding the requirements and mobilising various organisational members to drive its development. Yet, as the above discussion highlights, research on the individual antecedents on the social dynamics in software development is also characterised by conflicting findings. For example, software developers exert different expertise to perform various tasks, which thus demand their ability to communicate and network for diverse resources over different communication networks to coordinate tasks and drive performance. While some studies have attributed benefits to human factors others have played up a resources access stance instead as the main driver for software development success or failure. The interrelationship between the individual and organisational antecedents in software development also means that project activities are coordinated at the team and individual work structures.

For this, stream II details various layers of embeddedness in social interactions by which team members network and collaborate to access and manage diverse network ties across communication, relationship, and resource networks to ensure the success of the project. Thus, while different studies have considered individual embeddedness in offline and online networks as a potential for performance, other studies have also positioned that it is not just embeddedness across these media that matter but the actual nature of resources accessed and embedded in, that contributes to differing performance. Another strand of studies has also considered that access to such resources requires networking and embedding in both formal and informal work structures to drive performance. These positions reveal much about the multiplex nature of network ties possible that may be inseparable to truly make sense of their separate and overlapping embeddedness on performance advantages, which is currently lacking in software development research.

Finally, stream III describes the social dynamics of social networks, detailing the different aspects of network ties that vary in social and knowledge relations towards specific network outcomes. In summary, figure 5 provides a bird's-eye view to understanding and explaining why the multiplex network perspective is warranted to study the multilevel aspects of software developer structures. Since different natures of network ties can have bearing on potential outcomes. This is put into concrete and precise perspective as simplified in figure 4 and table 7 as our overarching lens to viewing software developer networks and performance.

2.6 Discussion of Results

Organisations function as socio-technical and transactive knowledge systems, whereby different tools, devices, knowledge, skills and attitudes of individuals and teams are combined to deliver organisational project goals (Cai et al., 2018; Cross et al., 2001; Leenders & Dolfsma, 2016; Sykes et al., 2014; Tang, 2015). Extant studies on software development scholars have confirmed the implications of social network structures on software development performance at individual, team and project level (Datta, 2018; Licorish & MacDonell, 2014; Lindsjørn et al., 2016; Nootjarat et al., 2015; Sawyer et al., 2010; Schmidt et al., 2014; Zhang & Venkatesh, 2013). However, there are limitations to these contributions based on their uniplex and limited multiplex perspective in studying the knowledge processes and network outcomes of individuals and teams coordinating software development tasks. A multiplex-based view (MBV) suggest that a multilevel perspective can drive existing research forward. Hence, a central question: *How can a multiplex-based view contribute to a software developer's coordination of resources towards performing software development tasks and optimising greater work outcomes?* To answer this requires setting two broad research objectives to focus on future research endeavours.

RO1: Investigate and explain how a multiplex perspective can enhance performance in software development settings.

Coordination is simply about linking different project requirements and dependencies to perform tasks and to accomplish project goals across formal and informal relations (Barthelmess, 2003; Ben-Menahem, Von Krogh, Erden, & Schneider, 2016; Cataldo &

Herbsleb, 2008; Kotlarsky, van Fenema, & Willcocks, 2008; Lindsjørn et al., 2016; Mishra et al., 2012; Nidumolu, 1995; Strode, Huff, Hope, & Link, 2012). Software development activity requires that organisations manage different parts of the development divisions (e.g., users, customers, software developers; general business, technical and technology requirements) at a formal level. Furthermore, software developers manage their tasks – partly formal and informal – to achieve personal, team and overall project performance. Individuals need to communicate and collaborate with skilful and knowledgeable others. Since individuals cannot rely on their knowledge alone, software developers will need to mobilise information and knowledge resources within their contacts (Anwar et al., 2017; Bernardi et al., 2018; Çaglayan & Bener, 2016; Kotlarsky et al., 2008; Park & Lee, 2017; Šmite et al., 2017). Studies reveal that the strategic embeddedness of software developers in the communication networks involving more informed and knowledgeable others (e.g., as gatekeepers) will increase their knowledge diversity and ability to perform technical and non-technical tasks (Cataldo & Herbsleb, 2008; Singh & Tan, 2010).

Cross, Borgatti, & Parker (2001) opine that employees seek multiple information and knowledge resources to solve problems, explore new sources, gain new perspectives and validate opinions through credible opinion leaders (e.g., resource brokers) to efficiently get work done (Helms & van Reijsen, 2008; Lee & Lee, 2015; Purvis & Zagenczyk, 2018). Because the majority of software tasks are handled by individuals, communication and collaboration among them will allow the sharing of diverse information and knowledge resources. Therefore, networking and establishing multiplex social ties is important for effective communication across formal and informal work structures. Although software development teams tend to be self-organised and managed, they are supported by senior project managers to control and keep project teams on track. In particular, keeping to projects' schedules and budgets, providing more information and knowledge support on technical adjustments, business and customer issues to ensure that projects are run smoothly (Ehrlich & Cataldo, 2014; Koivuaho & Laihonen, 2006; Lee, Park, & Lee, 2015; Lindsjørn et al., 2016; Nichols, 2006; Nidumolu, 1995).

Given the unstructured ways in which software developers access diverse information and knowledge resources across vertical and horizontal relationships (Jordan & Segelod, 2006), developing a multiplex communication profile may be warranted to foster performance. For example, earlier multiplex studies show that when an individual maintains simultaneously formal and informal relationships with other knowledge workers, innovative knowledge can be generated and performance enhanced (Aalbers et al., 2014; Cai et al., 2018). Similar research shows that when individuals maintain social ties to coworkers (i.e. "*horizontal, cross-unit ties")* to obtain knowledge related ideas, and also maintain ties to project managers (i.e. "*vertical, cross-hierarchy ties*") (Aalbers et al., 2016), such as idea connectors (Whelan et al., 2011); the individuals can gain buy-in from the management to further develop the knowledge ideas into innovation. Studies show that horizontal relationships permit teams to respond to business and technology changes, while vertical relationships permit teams to respond to changes in business and market technology (Chang et al., 2011).

Software development teams are expected to have a business and technical expertise, with the former relating to issues of features, functions, the in-house business process for systems environment, etc. The latter relates to issues of analysing user requirements, the feasibility of the technology, designing systems and configurations, programming, ensuring systems functioning and alignment of features (Lee et al., 2015; Park & Lee, 2017; Ryan & O'Connor, 2013). Again, multiple studies reveal that the overlapping of these two expert knowledge between two firms favours innovation performance (Belso-Martínez et al., 2017). Seeing related interaction effects on coordinating tasks, software innovativeness and performance will be valuable for research among software developers (Jordan & Segelod, 2006). Scholars suggest that social ties and good communication across offline and online networks can shape diverse access to and sharing of information and knowledge resources, developing quality software, great work satisfaction and improvement in individuals' and group performance (Du et al., 2007; Nidumolu, 1995; Park & Lee, 2014; Zhang & Venkatesh, 2013). Still, Peng et al., (2013) claim that it remains unclear which nature of network ties can optimally deliver high performance in software development. Research evidence suggests that not all social ties are centrally connected, so this can cause a disparity in their influence and impact on network outcomes (Heaney, 2014; Sasidharan et al., 2012; Sykes et al., 2014; Venkatraman & Lee, 2004). A trust multiplex network perspective can help channel more clarity around this puzzle.

Given that different avenues for information and knowledge sharing can influence performance, and their realisation can make such sharing effective, coordinating work through them across individuals, teams and organisations can be very significant (Ashworth & Carley, 2006; Du et al., 2007; Hossain & Zhu, 2009). For example, Kotlarsky et al., (2008, p. 99) suggest that "*coordination mechanisms become knowledge management instruments, with a focus on their contribution to the coherence of knowledge processes and activities in achieving coordinated outcomes*." As coordination mechanisms are considered complementary, it is argued that MBV can help capture the coherent nature of different dimensions that are embedded in software developer structures and knowledge processes to coordinate tasks. This includes what it might mean to apply a multilevel perspective to the study of social networks and implications for performance. It is thus paramount to understand what sort of social affiliations in formal and or informal relationship matter for multiplex ties, or relations in offline and/or online communication networks, or the diversity of information and knowledge resources (e.g., technical, business, problem solving, career, emotional support) being shared with respect to software tasks; whether all of these are requisite social foci that are linked to multiplexity in software development.

Research in this area should also provide an understanding of how different multilayers in social interactions impact the coordination of software development tasks towards optimising software project success. Thus, urging software development scholars to pay close attention to multiplex network structures in software development processes and related outcomes such as performance and innovation.

RO2: Understand how the nature of tasks and individual traits (e.g., personality) shape multiplex ties in software development settings.

"*The interplay between individuals' social interactions and traits has been studied extensively but traditionally from static or homogeneous social network perspectives*" (Meng et al., 2016, p. 32). Motivated by the question raised by Meng et al., "*to what extent does an individual shape the social networks the individual participates in and/or to what extent do the social networks shape the individual"* (Meng *et al.,* 2016, p. 32); scholars have researched into understanding how the nature of tasks and individual personality traits shape multiplex ties in software development settings. For instance, research suggests that work structure determines the nature of team structure (see for example Robillard and Robillard, 2000) and that knowledge work relies on experts' purposeful interactions among themselves (Kotlarsky et al., 2008). According to Robillard and Robillard (2000), the nature of collaboration varies in teamwork activities. These authors suggest that collaborative work can be *Mandatory*, based on formal regular routines and schedules; *Called*, based on two or more members agreeing to work unscheduled on a technical task; *Ad hoc*, based on two or more members working on the same task and sharing information and knowledge resources simultaneously; and lastly *Individual*, based on two or more team members performing unshared tasks with limited opportunity to interact together. These collaborative work activities may also be considered as low to a high level of work, resource, and cognitive cooperation. The combined effect of cognitive and workflows has been found to increase performance among software development teams (Zhuge, 2003).

Additionally, the increasing presence of social media and the real-time network linkage between individuals using smart devices, allow studying how individuals evolve in social network positions with multiple links and their traits (e.g., openness to experience, extraversion) (Meng et al., 2016) relevant for software development research. These personality traits are linked to starting communication, networking, exploring new knowledge environments, performing software tasks and driving project success (Acuña et al., 2009; Licorish & Macdonell, 2015; Strader et al., 2008). Figure 5 clearly shows that the interactions between individuals, activities, resources and medium of communication cannot be separated. Regarding that organisations and project developments provide an avenue for social interactions, the onus is on software developers to explore different networks available or needs to be created by them to achieve project goals. This implies that software developer personality traits can impact how he or she negotiates and network formally and informally with other developers across communication networks (offline, online) to access and utilise diverse resources (business, technical, etc.) towards performing individual and team tasks. For example, in informal work structures, individuals' social skills would be needed to gain network brokerage to stand a chance of being embedded in multiplex communication profile in stream II towards achieving the set goals in stream III.

Taken together, the nature of task interdependence, either high or low, varies across these collaborative activities, and overlapping interests or goals in different projects. This differential may perhaps shape the extent to which members form social ties to different work structures and interact with diverse information and knowledge resources towards coordinating software development activity (Jiang et al., 2019; Latorre & Suárez, 2017). Negoita et al., (2018) submit that different task interdependence: pooled, sequential, reciprocal and team led may affect whether team members work on a given task at the same time, or at different times, or need different or similar access to information and knowledge resources. That may deserve different coordination mechanisms to drive work and performance. Other studies show that the relation between task interdependence and developers' social structures can alter network performance (see for example Nan and Kumar, 2013). This study posits that such variations in the level of collaborative work requirements may however challenge the significance of multiplexity among software developers and IS research. Because research suggests that not all multiplex ties are relevant when considering their relative impact on resources flow (Venkatraman & Lee, 2004). Shipilov *et al.*, (2014) argue that researchers need to discern how different ties in one level directly or indirectly affect the other at another level and the ripple effect on the organisations. Conclusively, it is surmised that the significant application of multiplexity depends on social contexts, shared activities and issues of concerns between actors (Autry et al., 2014; Heaney, 2014; Mesch & Talmud, 2006). Thus seeking to understand those multiplex ties that are highly relevant and beneficial (Heaney, 2014) in software development settings is very crucial for future research. Regardless of any variations overall, multiplexity can offer phenomenal understanding to explaining how individuals mobilise their knowledge networks or social capital to acquire information and knowledge resources towards managing their work dependencies and improving performance in software development settings. Thus, making its significance undeniable.

Considering that MBV captures different dimensions of social dynamics in collaborative work, researching the role of individual personality traits of software developers in multiplex social networks and how this shape their performance would be worthwhile since extant studies on personality show how individuals differ in their behavioural characteristics. For example, some people might be extroverts, introverts, sensing, intuitive and thinkers (Strader et al., 2008). Personality traits also account for why some software developers occupy central positions to code and diffuse knowledge (Licorish & Macdonell, 2015), communicate, form teams, generate ideas, drive software quality (Acuña et al., 2009; Acuña et al., 2015) and increase IS proficiency (Kane & Borgatti, 2011). Also, this has enabled individuals to network and negotiate for diverse social ties to access richer information and knowledge resources to influence performance (Mehra et al., 2001; Sasovova et al., 2010). For example, a study by Mehra et al. (2001) reveals that self-monitoring personality (i.e. an individual's ability to modify/adapt their social behaviour), in particular, high self-monitors act as go-betweenness in social networks, and can bridge structural holes to access and control the flow of diverse resources. This study suggests that high self-monitors can attain strategic position and then translate it to high performance than low self-monitors because of their networking social skills (Sasovova et al., 2010). While this study focused on dyadic relationships, a multiplex relationship study in software development environments will be valuable. As delineated in figure 5, it would be more valuable to further understand whether it is the personality trait, task structure, team structure and communication structure that tends to affect an individual's opportunity to attain network positions and bridge for resources in multiple networks. This may extend insight into structural holes theory or brokerage literature. Obstfeld et al., (2014, p. 139) claim that brokerage potentials demand having ties to two or more individuals. Similarly, individuals may involve in multiplex ties in parallel, as evident in multiplex studies. The brokerage literature reveals that a resource broker could access resources in one group and present them for use by another group (Chung & Jackson, 2013; Li et al., 2018). This study argues that such resources may not only enhance the actor's performance in work-related activities but also help their team gain performance through effective utilisation of those resources (i.e. conduit brokerage) (Obstfeld et al., 2014).

To this end, examining the interaction between software development tasks and software developer personality traits in multiplex networks and performance is therefore of research interest. Scholars claim that few studies have observed the relation between multiplex structures and individual traits (Meng et al., 2016). Tasks are often divided independently among software developers, and where interdependence is necessary, communication mechanisms are needed to achieve coordinated software development (Cataldo & Herbsleb, 2008; Nichols, 2006). Hence individual software developers would be expected to mobilise their social networks to facilitate performing work efficiently and effectively. How individuals can achieve this is partly subject to their characteristics such as personality traits in the communication networks (Cataldo & Herbsleb, 2008; Rasch & Tosi, 1992). Synchronising work activities through work structures and engaging in boundary-spanning for external resources are considered to be effective strategies for coordinating work demands in software development projects (Strode et al., 2012).

Other pertinent questions that future research should consider are:

- What types of advice or knowledge networks, and positions within those networks, are most pertinent in explaining software developer performance?
- How should online and offline networks be structured to enable high performance in software development?

2.7 Conclusion

This paper reviews two decades of research on network multiplexity, relating this to the complex process of software development and its performance consequences. It considers the literature on the antecedents of network embeddedness, addresses the dynamic nature of networks relative to multiplexity and discusses some of the mechanisms through which intra-organisational ties become multiplex in a software development context. Overall, it details how the lens of multiplex social networks can increase our understanding of performance in software development settings. This study identifies that while valid inroads have been made into the phenomenon of network multiplexity in an IS context, significant gaps and unresolved questions remain. These are particularly attributable to separate literature, that warrants current merging in the near future the literature on intra-organisational networks and the IS literature on software development.

For instance, extant studies have shown that the coordination of software development activity requires a different level of social interactions through which software development team members network, communicate and collaborate to access diverse information and knowledge resources to perform software development tasks. Since coordinating resource mechanisms occur at individual, team and organisation level, an understanding of the different structural dynamics that guides such work coordination to perform software development tasks, as well as implications for performance, have been of interest to many IS scholars. Social network analysis (SNA) has been applied to understand the effect of social interactions on software development network outcomes. This includes the effect of software developers' network positions in mobilising social capital and resources access, which has also been linked to accessing diverse information and knowledge resources, software development performance, individual and team performance, team cohesion and systems use proficiency. However, these studies which apply SNA to study people's social networks and resource flow among software developers' communications are limited. Furthermore, they almost exclusively considered uniplex social ties due to the neglect of the embedded nature of real-world network relationships. This study proposes that a multiplex-based view (MBV) can help scholars drive existing research contributions forward towards understanding the dynamics inherent in these work structures (i.e., software developer networks), and how their performance implications can be better understood.

In sum, this study posits that software development researchers can benefit from the inherent potentials of a multiplex view to better understand and explain how structural connections over different layers of social ties in communication networks deserve attention. This among other reasons is why MBV matters to offering deeper insights on what multiplexity might mean for software developers' coordination of software development activities. Drawing insights from structural configurations and resource brokering should provide an understanding of how offline and online communication networks are socially constructed, and how cross-cutting ties in the multiplex social networks guide knowledge management processes. This can also moderate the methodological deficits that may have existed because of such research limitations. For example, social network research has predominantly been driven by quantitative data and less adoption of the mixed methods approach (i.e., quantitative/qualitative data). In addition to adopting more of longitudinal and not mainly cross-sectional design to better understand how multiplex ties evolve or can dissolve, and what this might also mean for attaining and sustaining performance over a timeline, as highlighted in Table 7 (Kane et al., 2014; Whelan et al., 2016).

Considering burgeoning interests and the wide application of multiplexity in broader management science areas, this seems a plausible path forward for management science and IS scholars to appreciate a more holistic view of multiplex social ties, thus extending the body of knowledge on how social networks have been operationalised in software development and general management fields. To conclude, Filiposka *et al.*, (2017, p. 835) suggest that "*by approaching the problem of different interrelationships between actors as a multiplex network problem, one can gain useful insight on the importance of each type of link within the social network as a whole, as well as on the interaction and overlap between different link types, especially in the offline/online* *(real/virtual) environments*." In other words, multiplex social ties can help observe how individuals manage different communication profiles in accessing, utilising, and optimising resources and improving performance. This complimentary portfolio of network ties in the communication networks matters for coordinating tasks and driving performance among individuals and teams in software development and general project management.

Chapter 3 A Study of Offline and Online Knowledge Brokerage among Software Developers

3.1 Introduction and Chapter Outline

This chapter is built on the social network dimensions of multiplexity observed in the previous chapter by looking specifically at the dynamics of brokerage behaviour of software developers and the nature and type of knowledge shared on their software development effort. Section 3.2 describes the abstract of this study. Section 3.3 introduces the research background. Section 3.4 provides the literature review of network brokerage. Section 3.5 details the research method in carrying out the research. Section 3.5 describes the research results. Section 3.6 provides discussions and conclusions of this study. Finally, section 3.7 concludes and presents the research limitations and suggests future research.

3.2 Abstract

Access to the knowledge needed to make better decisions, enhance performance, and develop innovation does not exist in a vacuum. Individuals and technologies play a crucial role to ensure people get access to useful knowledge. The mediating role of brokers has been found to be important in connecting otherwise disconnected individuals to pertinent knowledge resources through offline and online communication networks. While studies have considered both business and technical knowledge as complementary knowledge resources needed in developing software, there is a lack of sufficient understanding of how this knowledge is sourced separately and simultaneously in offline and online communication networks. To understand the communication roles of brokers in accessing and disseminating knowledge, a case study was conducted in a software development unit, which resulted in gathering both social network and interview data. The findings revealed varying patterns of brokers and communication roles emerged across these knowledge and communication networks. The implications in offline versus online brokerage and areas for further research endeavours are discussed.

***Keywords*:** Software development, Social network, Brokerage, Communication network, Social network analysis, Software developer, Offline brokerage, Online brokerage, Business domain knowledge, Technical knowledge

3.3 Introduction

The COVID-19 pandemic has made us more aware of the important role of technologies in facilitating social interaction, in addition to the mediating role of individuals in granting access to information and knowledge resources (Ågerfalk, Conboy, & Myers, 2020; Waizenegger, McKenna, Cai, & Bendz, 2020; Wu, Antone, Srinivas, DeChurch, & Contractor, 2021; Yang, Soltis, Ross, & Labianca, 2021). Based on social network tradition, communication networks allow individuals and groups to share diverse ideas, information, and knowledge resources. Since the knowledge needed to make better decisions, enhance performance, and stimulate innovation does not exist in a vacuum. Individuals and technologies now play a crucial role in ensuring that people get access to the knowledge they need in offline (face-to-face) and online (email, instant messaging, electronic whiteboards, video conferencing) workplace communication networks, which has been linked to varying performance outcomes (Ding et al., 2019; Rico & Cohen, 2005; Zhang & Venkatesh, 2013; Zhang et al., 2008). The presence of a broker in a unit has been shown to enhance the performance of the unit and those connected to them (Allen, 1984; Balkundi, Harrison, Balkundi, & Harrison, 2006; Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014; Havakhor, Soror, & Sabherwal, 2018; Sawyer et al., 2010). This includes expanding knowledge and financial resources (Bilecen & Cardona, 2018; Reus et al., 2020), and developing innovation (Belso-Martínez et al., 2017; Kauffeld-Monz & Fritsch, 2013; Lam, 2018; Yuan Li et al., 2018; Whelan et al., 2011).

In software development, the nature of work is knowledge-intensive because it is complex and requires a creative approach to solving a variety of problems (Boden & Avram, 2009; Dissanayake, Dantu, & Nerur, 2013; Gallivan, 2003; Payne, Keith, Babb, & Spruill, 2018; Tiwana, 2004; Wang, Huang, & Yang, 2012). A software developer's knowledge alone may be insufficient to tackle arising task problems and would need to rely on the knowledge and social skills of other individuals who are more skilful, knowledgeable, experienced, and perhaps, more influential also to mobilize such performance-enhancing resources to improve work efficiency and effectiveness (Park & Lee, 2017; Šmite et al., 2017). Software development scholars have found business domain and technical knowledge as two important knowledge needed to be integrated into developing software (Mehta & Bharadwaj, 2015; Ryan & O'Connor, 2013; Šmite et al., 2017; Tiwana, 2004, 2012). These scholars argue that each type of this knowledge allows software developers to meet different requirements in software development. It is differentiated that business domain knowledge relates to knowledge in understanding the business processes and rules, system support and functions, and design objectives of customer's business needs. Whereas technical knowledge relates to knowledge in understanding design and constraints, development methodology and tools, testing and debugging codes, programming language, and coding environment of technological needs. Empirically, Tiwana (2004) finds that integrating both business domain and technical knowledge decreases software defects and density but increases its effectiveness and efficiency during the software development process. Tiwana (2012) and Vermerris, Mocker, & Van Heck (2014) have argued that although the use of business domain knowledge and technical knowledge are complementary in software development, they tend to be communicated (and studied) separately, which is problematic in timely accessing and integrating them across project functions by software development teams. Surprisingly, the structure of knowledge sharing network by which software developers get access to these two types of knowledge combined in integrating them to perform software development tasks tends to be understudied, thus the need for extended studies (Anwar et al., 2017; Boden & Avram, 2009; Cataldo & Herbsleb, 2008; Mehta & Bharadwaj, 2015; Ryan & O'Connor, 2013; Šmite et al., 2017; Tiwana, 2004, 2012; Vermerris et al., 2014).

Additionally, similar studies have investigated the roles of individual embeddedness and brokering activity in different network contexts online (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014; Havakhor et al., 2018; Reus et al., 2020; Sowe, Stamelos, & Angelis, 2006) and offline (Aalbers & Dolfsma, 2015; Balland et al., 2016; Belso-Martínez et al., 2017; Boden & Avram, 2009; Burt, 2004; Cross, Borgatti, & Parker, 2001; Kirkels & Duysters, 2010; Lam, 2018; Mehta & Bharadwaj, 2015; Sawyer et al., 2010; Whelan et al., 2011; Whelan, Teigland, Donnellan, & Golden, 2010; Whelan, Golden, & Donnellan, 2013) communication networks on different organisational outcomes (e.g., individual performance, innovation development). Still, none of these studies has inquired to understand the dynamic nature of brokerage roles in different knowledge interactions, and across offline and online communication networks simultaneously. In filling the above gap, the research question asks: *How do online and offline knowledge brokering activities contribute to software development?* The objective of this study is to understand the way different networking behaviour of brokers, brokerage (communication) roles, and the role of knowledge flowing through their networks contribute to the software development effort. This study investigates this question and contributes to a better understanding of the diversity of business domain and technical knowledge sharing in face-to-face offline and online brokerage in informal intraorganisational networks (Aalbers & Dolfsma, 2015; Moser et al., 2017; Reus et al., 2020).

The findings revealed varying patterns of brokers and communication roles (e.g., coordinator, gatekeeper, representative, consultant) that have emerged, and are pertinent in software development knowledge sharing across the communication networks. Knowledge brokering activities are active in both business domain and technical knowledge sharing in face-to-face offline brokerage, and partly in technical knowledge sharing but none in business domain knowledge sharing in online brokerage. The lack of online brokerage in business knowledge sharing is surprising but provides more opportunity to focus on cultivating online (beyond offline) brokers as people now work remotely due to the COVID-19 pandemic. Overall, the result revealed that individuals maintaining both internal and external knowledge orientation in either offline brokerage or online brokerage will contribute to the software development efforts. These findings generate insights about the role of knowledge in brokerage processes of software developers in both offline and online communication networks simultaneously. It also serves as a window through which individuals engage in coordinated action for network performance

This study takes a step further to observe the positioning and functioning of brokers across these media simultaneously, and what this might mean for cultivating brokers to function more effectively in the new norm of the remote working environment. The COVID-19 crisis has caused a radical shift in rethinking organisational work routines from workplace team-based collaboration to working individually and remotely from home, including software development teams. Thus, presenting an interesting area for extending research contributions (Ågerfalk et al., 2020; Greve, 2020; O Connor, Conboy, & Dennehy, 2021; Waizenegger et al., 2020).

The remainder of the paper is organised as follows. Section 2 presents an overview of the literature. Section 3 describes the research setting, the sample, its construction, and the network methods used to analyse it. Section 4 presents the results. Section 5 sets out the key findings, contributions, research limitations, and avenues for future research.

3.4 Literature Review

3.4.1 Brokers in Communication Networks

Networks are dynamic (Chen et al., 2021). This dynamism stems from the fact that people might choose to network or not with other individuals and groups for collaboration contingent on actors' strategic actions (Ahuja et al., 2012; Burt et al., 2016; Kleinbaum, 2016; Montanari et al., 2016; Pollock, Porac, & Wade, 2004), economic survival instincts (Desmond, Murphy, Slez, & Lo, 2012), opportunity, status, and power (Labun, Wittek, & Steglich, 2016; Oh, Charlier, Mount, & Berry, 2014; Smith, Menon, & Thompson, 2012), learn and gain new knowledge (Cross et al., 2001; Peters et al., 2010). At the centre of network structure lies the issue of relationships and linkage access to information and knowledge resources about executing tasks among individuals in their communication networks, thus partly influencing the attention given to brokers in different fields (Haas, 2015; Halevy, Halali, & Zlatev, 2019; Kwon, Rondi, Levin, De Massis, & Brass, 2020; Long, Cunningham, & Braithwaite, 2013; Wellman, 1983). For these reasons, scholars have considered the theories of self-interest (e.g., structural holes), collective action, and co-evolution theories (e.g., multiplexity) to account for different levels of network linkages and affiliations among individuals, groups, and organisations (Contractor & Monge, 2003). Burt (2004) posits that structural holes refer to gaps in network positions between otherwise disconnected individuals in network structures. This position has tended to serve as a mechanism to access alternative sources of good ideas, information and knowledge resources, divergent perspectives, etc. Structural holes are more focused on the positional advantage of the middle person. Meanwhile, multiplexity theory refers to two individuals engaging in multiple resource relations simultaneously (Lazega & Pattison, 1999; Verbrugge, 1979). Other empirical studies have shown that brokerage of structural holes has led to generating information quality (Sasidharan et al., 2012), while multiplexity has led to generating innovative knowledge in workplace settings (Aalbers, Dolfsma, & Koppius, 2014). This study pays attention to these theories.

Viewing brokerage as a central concept to understand social network exchange, social network scholars have regarded brokers as individuals that bridge otherwise disconnected pockets of knowledge. Within an organisational setting, this implies individuals whose roles include creating and facilitating information and knowledge resource connections between two or more persons or organisations with no cognitive, cultural, and physical ties, or for lack of trust by sharing knowledge resources with them (Balkundi et al., 2006; Kadushin, 2002; Lam, 2018; Long et al., 2013; Marsden, 1981). While this is so in a traditional view of brokerage, Obstfeld, Borgatti, & Davis (2014) claim that a brokerage may not necessarily mean the presence or absence of connections, generically structural holes, but the potential of brokers to be connected to two or more social actors. Taking into cognisance the vitality of having ties to connected and unconnected individuals (i.e., direct, and indirect contacts). Brokering activity involving unconnected others is what other scholars have described as absolute brokerage (Yuan Li et al., 2018).

Scholars suggest that more often network brokers tend to emerge instead of being assigned to act in knowledge broking roles inside and outside their organisations (Boari & Riboldazzi, 2014; Esteve Del Valle & Borge Bravo, 2018; Kislov, Hodgson, & Boaden, 2016; Schlierf & Meyer, 2013; Whelan et al., 2011). Individuals occupying brokering roles have been characterised as boundary spanners, gatekeepers, representatives, consultants, coordinators, liaisons, or itinerants (Aalbers & Dolfsma, 2015; Boari & Riboldazzi, 2014; Boden & Avram, 2009; Gould & Fernandez, 1989; Kirkels & Duysters, 2010; Whelan et al., 2013), network integrators, idea scouts, idea connectors (Parise, Cross, & Davenport, 2006; Whelan et al., 2011), knowledge managers, capacity builders, linkage agents (Meyer, 2010), strategic network architects (Pollock et al., 2004) and simply resource brokers (Mortensen & Haas, 2018). While these brokerage roles can be differentiated, it is however acknowledged that individuals can play up two or more roles such as acting as brokers and knowledge providers (Sowe et al., 2006), as idea scouts and idea connectors in organisations (Whelan et al., 2011). In sum, brokers are deemed "able to link know-how, know-why, and know-who" across network boundaries (Meyer, 2010, p. 119).

Therefore, the variety of functions of brokers includes identifying and localising knowledge, redistributing, disseminating, rescaling, and transforming knowledge in their networks (Schlierf & Meyer, 2013). Lam (2018) asserts that brokering activity brings transformation to brokers and their work environment. A big part of this is the intermediation of knowledge as the need to connect indirect contacts and put ideas on the table for discussion becomes essential in helping others to execute work tasks. While brokers are often not the producers of knowledge, they tend to be a driving force in facilitating processes and coordinating tasks in their work environment (Kauffeld-Monz & Fritsch, 2013; Schlierf & Meyer, 2013). Kauffeld-Monz & Fritsch (2013) suggest that an unconnected network tends to have more brokerage positions, which may result in many ties, hence more collaborations with many others. While bemoaning that a broker might be carefree of maintaining knowledge advantage based on knowledge sharing with others. Rightly so because a broker can share either tertius iungens ("the third who joins") or tertius gaudens (“the third who enjoys”) orientation (Obstfeld, 2005; Obstfeld et al., 2014), recently termed tertius separans (“the third who separates”) in brokering activities (Kwon et al., 2020). In-between Tertius iungens and Gaudens is the one who mediates network contacts (conduit). In tertius iungens orientation, a broker is more interested in bringing actors together to facilitate information and knowledge resources between them, with the option to either stay or step aside in this brokerage. In contrast, in tertius gaudens or separans orientation, while a broker is interested in facilitating information and knowledge resources, he or she also delights to act as an opportunist in taking advantage of their separation and so not willing to join them (Grosser, Obstfeld, Labianca, & Borgatti, 2019; Kadushin, 2002; Obstfeld, 2005; Obstfeld et al., 2014), which would be important in fostering communication, collaboration and coordination of tasks together within and between teams. Particularly for a broker who shares multiple resources with a pair of other actors to be a super-connector in the communication networks (*multiplex ties*).

Finally, and to advance insights on these brokerage orientations, Grosser et al. (2019) classified brokerage orientations into Tertius iungens orientation and Disjoint brokerage orientation. A brokerage orientation is when an actor exhibits and performs a specific type of brokerage. Tertius iungens orientation is further divided into Brief iungens and Sustained Iungens. Meanwhile, disjoint brokerage orientation is also divided into mediation and separation. The below figure 6 describes this brokerage typology.

Brokerage Category	Brokerage Type	Description
Tertius Iungens Orientation	Brief *Iungens*	To introduce or facilitate preexisting ties between parties such that the coordinative role of *tertius iungens* subsequently recedes in importance.
	Sustained *Iungens*	To introduce or facilitate interaction between parties while maintaining an essential coordinative role over time.
Disjunct Brokerage Orientation	Mediation	To intermediate between parties who cannot, or prefer not to, interact with one another. The mediation broker is a kind of *non-partisan* (Simmel, 1950) that is similar to the conduit broker (Obstfeld, Borgatti, & Davis, 2014).
	Separation	To see advantage in having disconnected alters and to actively work to maintain separation among network contacts. This orientation reflects Simmel's (1950) conception of active *tertius gaudens* brokerage.

Figure 6. Brokerage Orientation Typology (Grosser et al., 2019, p. 47)

Obstfeld et al. (2014) argue however that maintaining and leveraging structural advantages often come with difficulty for brokers. Except when such relationship between the brokers and others (alters) are multiplex (Kwon et al., 2020; Podolny & Baron, 1997). Studies suggest that capitalising on more rather than fewer structural holes can drive better performance (Burt & Soda, 2021). Although Grosser et al. (2019) contend that structural holes within any network structure might have offered resource benefits, the ability to exploit them depends on an individual's brokerage action to achieve desired benefits.

3.4.2 Online versus Offline Brokerage

Communication networks consist of social relationships where individuals share information and knowledge resources in both offline and online networks (Cataldo & Herbsleb, 2008; Dennis et al., 2008; Zhang & Venkatesh, 2013; Zhang et al., 2008). In this context, an *offline brokerage* describes a brokerage process whereby a broker connects two individuals within and between groups with information and knowledge in need to perform tasks in a face-to-face offline network. An *online brokerage* is a brokerage process whereby a broker connects two individuals within and between groups with information and knowledge in need to perform tasks in the online network. To reiterate, research on brokers has tended to focus on face-to-face, offline networks, or online networks, but relatively little on how the two combine in network studies (Kadushin, 2002). In software development, communication networks serve as a good mechanism for software developers to know who is doing what, how it can be done, and the lead (and cycle) times to get things done (Cataldo & Herbsleb, 2008; Mishra et al., 2012; O Connor et al., 2021). Both offline and online networks are also important to move information and knowledge around among direct and indirect contacts to meet similar and different needs (Haythornthwaite, 2001, 2002; Zhang & Venkatesh, 2013). The role of digital technologies has made the online network a great potential for storing and retrieving documents, generating, facilitating, and diffusing valuable information and knowledge resources with a large number of collaborators through multiple (and enterprise) social networks simultaneously, while also blending it with the face-to-face offline network with work colleagues, customers, and friends (Reus et al., 2020; Verbeke, Dietz, & Verwaal, 2010; Verona, Prandelli, & Sawhney, 2006; Whelan et al., 2013).

Reus et al. (2020) and Verona et al. (2006) suggest that the distance and easiness of an online network make it preferable to a face-to-face offline network. Arguing that a broker can easily access, switch, link, and distribute knowledge to a diversity of groups in the online network. Because it tends to provide greater flexibility and cost less to connect with many groups and collaborators at the same time from different geographical locations. The benefits of reaching out to a larger audience make the online environment more widely acceptable to create direct and indirect ties. The use of technologies is inextricably inevitable given the current COVID-19 that has compelled organisations and their knowledge workers to adapt to the rapidly changing work environment from office settings to conforming unusually to working from home. The need for adaptability has forced knowledge workers working remotely, to make changes to their social contacts in their interpersonal networks by activating past ties and forging new ones to manage stress and achieve performance goals (Giorgio, Mascia, & Cicchetti, 2021; Wu et al., 2021). Overall, the use of the online network to engage, collaborate and coordinate work activities among knowledge workers have become an evolving working culture that is pivotal in promoting organisational knowledge sharing among workers (Ågerfalk et al., 2020; Greve, 2020; O Connor et al., 2021; Rico & Cohen, 2005; Waizenegger et al., 2020). There is a general concern that offline network channels have more information quality and physical appeal to senses than the online network during interpersonal communication between individuals. Considering the importance of social ties, individuals maintaining direct ties tends to have strong ties, and have been found to enable easy sharing of complex tacit knowledge. However, individuals maintaining indirect ties tend to have weak ties but are more suitable for sharing codified knowledge. In doing so, weak ties serve as conduits of learning and new knowledge sources. In contrast, it is easy for individuals to share redundant knowledge with strong ties (Granovetter, 1973; Levin, Cross, & Abrams, 2002; Rico & Cohen, 2005; Verona et al., 2006; Zhang & Venkatesh, 2013). Notwithstanding scholars assert that redundant knowledge is useful in better interpretation, building and preserving groups' identities (Reus et al., 2020). Overall, Zhang & Venkatesh (2013, p. 702) posit that "*Online and offline workplace communication networks enable access to diverse sources, such as job-related knowledge, task advice, strategic information, and social support.*"

The vital role of brokers in offline and online networks often becomes evident when they exit and the network fragments as result. This may be costly for facilitating, disseminating, and integrating knowledge resources in coordinating work practices more effectively (Everett & Valente, 2016; Parise et al., 2006). These networks are considered paramount to leading a successful open source or closed source software development (Bird, Gourley, Devanbu, Gertz, & Swaminathan, 2006). For example, in an online communication network study, using social network analysis, Sowe et al. (2006) find that brokers can close the knowledge gap between expert software developers and community users, while also extracting and managing the value of software knowledge, and facilitating shared learning experience with community users in open-source software development. Additional software development studies have found that brokerage roles of technical leaders or technical experts have been instrumental in effectively sharing more information and extending productivity among teams in their networks (Ehrlich & Cataldo, 2014; Šmite et al., 2017). In online enterprise social networks, a broker's function has also brought about the addition of new knowledge to expand the group's knowledge base (Reus et al., 2020). Therefore, brokers in online networks benefit from the potential of technologies to secure and maintain direct and indirect network ties in accessing, generating, and disseminating relevant information during the innovation process with online community users without the time and geographical barriers (Verona et al., 2006). Such roles can be related to technological gatekeeping observed within research and development networks on how brokers can be positioned to tap the benefits of online technologies in mobilising informational resources for innovation development in organisations (Whelan et al., 2013, 2010). In doing so, organisations can build on onsite direct engagement and knowledge resourcefulness.

Moreover, and in an offline communication network, studies have shown how brokerage patterns in social interactions explain performance in software development (Sawyer et al., 2010). Hence, it is more important for brokers to network across social structures to motivate social practices (Akkerman & Bakker, 2011; Obstfeld et al., 2014) either for their safety or effectiveness in their network (Kadushin, 2002). It is claimed that effective brokers must have good interpersonal, communication, and motivational skills (Cvitanovic, Cunningham, Dowd, Howden, & van Putten, 2017), as well as the cognitive ability to manage information, absorb, codify, and share tacit knowledge with people. While also working cooperatively with them by facilitating shared learning and demonstrating leadership on how such knowledge can be utilised to drive solutions to work problems (Conklin, Stolee, Harris, & Lusk, 2013; Cvitanovic et al., 2017; Kislov et al., 2016; Lam, 2018; Reus et al., 2020; Verbeke et al., 2010). Similar offline-based brokerage studies in non-software development settings also show why organisational embeddedness and leverage of structural holes in both business and technical networks have resulted in greater innovation outcomes (Belso-Martínez et al., 2017).

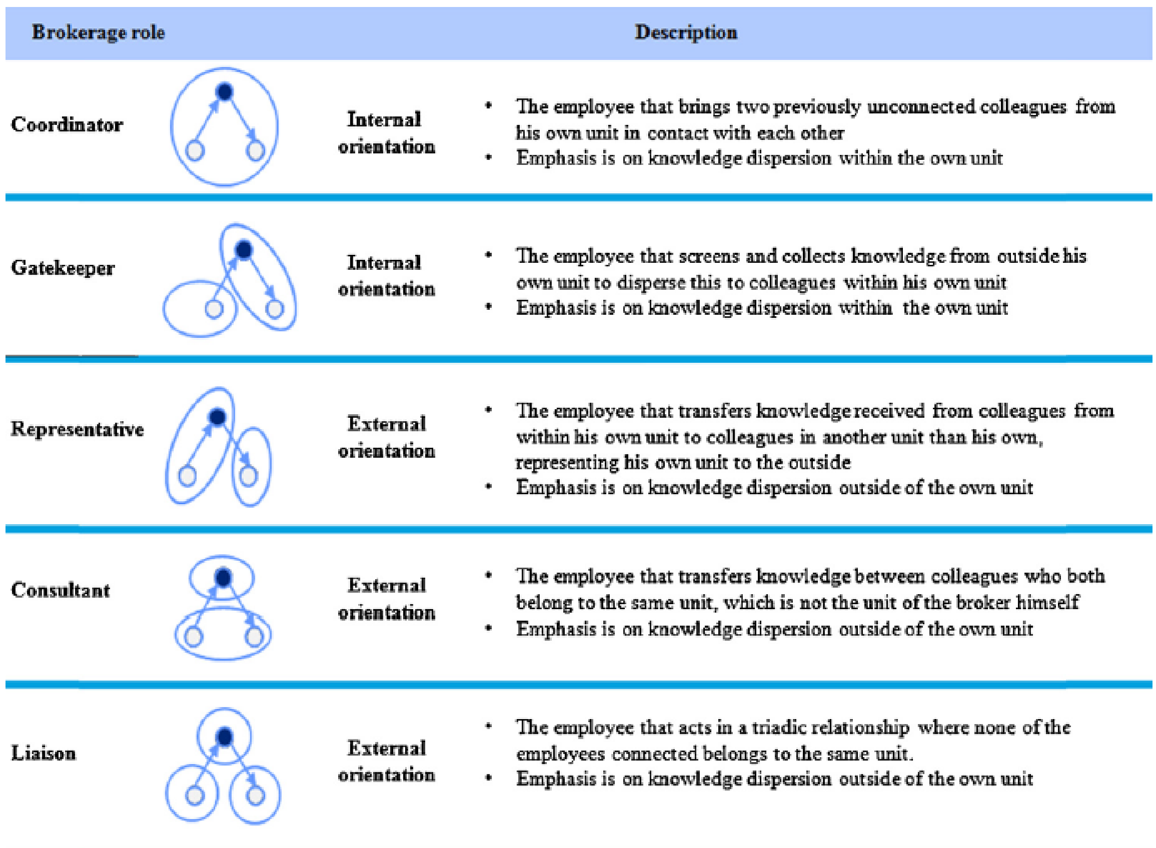
3.4.3 Determining Brokerage Roles

Fundamentally, social network analysts have found that brokerage positions of individuals can be determined based on their network centrality measures: in-degree, out-degree, and betweenness centrality (Bird et al., 2006; Esteve Del Valle & Borge Bravo, 2018; Everett & Valente, 2016; Freeman, 1978; Kauffeld-Monz & Fritsch, 2013; Sasidharan et al., 2012; Winch & Courtney, 2007). Others have considered other forms of brokerage orientations: coordinating, gatekeeping, representing, consulting, and liaising (Aalbers & Dolfsma, 2015; Boari & Riboldazzi, 2014; Gould & Fernandez, 1989; Kirkels & Duysters, 2010). The focus is on betweenness centrality and these other differentiated brokerage roles to provide a better explanation in software development settings. These measures help in understanding brokerage processes and actions that drive software development success.

Social network analysis (SNA) is a technique for identifying and understanding the position of employees, their relationships, and interactions within communication networks. It helps to know how a broker functions either effectively or not within their community of practice. A degree is simply the number of ties a person has based on their position within or across a network and shows how connected the person is. It is categorised into out-degree and in-degree centrality. *Out-degree* describes the number of social contacts a person goes to for advice, information or knowledge resources. In-degree centrality describes the number of social contacts a person provide resources to (popularly described as in-degree centrality). It is also linked to the frequency of communicational activity. A betweenness centrality simply relates to an intermediary position that a person occupies between two unconnected others (structural holes) that permit such a person to source and share non-redundant information and knowledge resources. An individual that fulfils a strong brokerage position will be considered as either a broker, or most often a gatekeeper in social networks, and such position has been linked to achieving high status, power, and control of information and knowledge flow – which can also create a bottleneck in a network of brokers when removed or prevent the passage of information and knowledge to others (Bird et al., 2006; Everett & Valente, 2016; Xiao & Tsui, 2007).

Kauffeld-Monz & Fritsch (2013) posit that betweenness centrality considers the importance of the indirect ties in the network, and thus indicate the extent to which brokers can absorb and share the factual or declarative type of knowledge (know-what), and scientific knowledge (know-why). This includes being key sources of novel ideas during the innovation process (Bird et al., 2006; Winch & Courtney, 2007). For instance, brokers acting as "go-betweenness" in-network centrality have been found to generate quality information and achieve greater work effectiveness. This is why betweenness centrality is equally acceptable by network scholars as a good measure of structural holes (Sasidharan et al., 2012). It is therefore suggested that the presence of at least one broker should improve knowledge sharing within their network (Kauffeld-Monz & Fritsch, 2013). The convergence of roles can also lead to an overwhelming of responsibilities to be less an effective broker if not well supported by others and their organisations (Burt & Soda, 2021; Long et al., 2013), particularly in a collectivist, not individualist culture (Xiao & Tsui, 2007).

Despite the importance of betweenness centrality proposed by Freeman (1978) to establish brokers' positions, it is argued that centrality measures fail to differentiate differing brokers' (communication) roles (Kirkels & Duysters, 2010); which is addressed by Gould & Fernandez's (1989) brokerage measures. These authors classified the communication roles of brokers as coordinator, gatekeeper, representative, consultant, and liaison or itinerant (see Figure 7). They argue that a broker might perform any of these roles combined. A coordinator functions by connecting two disconnected others with knowledge resources in their unit. A gatekeeper functions by sourcing knowledge from outside to be shared with members in their unit. A representative functions by representing their unit in sharing knowledge with members of another unit. A consultant functions by sharing knowledge with members of the same unit other than their unit. A liaison or an itinerant functions by sharing knowledge in a triadic relationship with members of different units other than their unit (Aalbers & Dolfsma, 2015; Boari & Riboldazzi, 2014; Kirkels & Duysters, 2010). The role of coordinator and gatekeeper shared an internal orientation, whereas the role of representative, consultant, and liaison shared an external orientation in brokerage orientation. For instance, Aalbers & Dolfsma (2015) have found that individuals who shared both internal and external orientation were more successful at bridging firm-internal boundaries to drive innovative knowledge transfer. Ideally, the essence of brokers in communication networks would be to pave way for open communication, enabling socialising and sharing of knowledge, fostering collaboration, and promoting shared understanding to coordinate tasks and achieve performance (Boden & Avram, 2009). In this context, betweenness centrality and brokerage roles were adopted to capture the network behaviour of software developers.



Brokerage role		Description
Coordinator	Internal orientation	• The employee that brings two previously unconnected colleagues from his own unit in contact with each other • Emphasis is on knowledge dispersion within the own unit
Gatekeeper	Internal orientation	• The employee that screens and collects knowledge from outside his own unit to disperse this to colleagues within his own unit • Emphasis is on knowledge dispersion within the own unit
Representative	External orientation	• The employee that transfers knowledge received from colleagues from within his own unit to colleagues in another unit than his own, representing his own unit to the outside • Emphasis is on knowledge dispersion outside of the own unit
Consultant	External orientation	• The employee that transfers knowledge between colleagues who both belong to the same unit, which is not the unit of the broker himself • Emphasis is on knowledge dispersion outside of the own unit
Liaison	External orientation	• The employee that acts in a triadic relationship where none of the employees connected belongs to the same unit. • Emphasis is on knowledge dispersion outside of the own unit

Fig. 1. Communication roles. *Note*: Adapted from Gould and Fernandez (1989). Dots indicate individuals, circles indicate business units, and arrows indicate the flow of information and resources between individuals.

Figure 7. Brokerage communication roles (Aalbers & Dolfsma, 2015, p. 101)

3.5 Research Method

3.5.1 Research Setting

To understand the dynamics of knowledge brokerage processes in software development settings, this study is focused on the role of brokers in the business and technical knowledge sharing in offline and online networks of software developers. The setting is a software-oriented organisation and has requested their name to be anonymous. It is a Fortune 500 company that develops computer networking services. The organisation is headquartered in the USA and had annual revenue of $129 billion in 2019. It employs approximately 61,000 globally, including 700 people employed across 3 locations in Ireland. This data was gathered from one of these sites in Ireland. The work activities of this site focus on developing and releasing software code into production for end users. This includes developing and automating delivery applications into the cloud, automating the development tools for systems applications and integrations, designing solutions to implement, maintaining systems infrastructure, and providing support for various products to external users. These activities related to software development provide an avenue to consider investigating the structure of knowledge brokering activities among software developers.

3.5.2 Research Process and Data Collection

The data gathering process is divided into two phases. Phase 1 involved preliminary interviews to understand the pertinent knowledge interactions by software developers. Phase 2 involved a follow up web-based questionnaire design based on interviews conducted. The below provides further details of each phase of the research process.

3.5.2.1 Phase 1 – Pilot Interviews

Firstly, the case setting deserved an exploratory study to explore the reality of knowledge interactions and network ties being experienced among software developers, using unstructured face-to-face interviews, in line with the dramaturgical model in qualitative approaches to the case study (Antwi & Kasim, 2015; Beissel-Durrant, 2010; Myers & Newman, 2007). This is aimed at generating insight, interpreting, describing, and building on the theory being studied (Eisenhardt, 1989; Yin, 1994). Based on individuals' permission to participate and record their interview, the exploratory interviews lasted on the average length of 45 minutes with 6 experienced (11-40, in average years) software developers to gather insight into the nature of their work roles, which revealed that business domain and technical knowledge are pertinent resources needed to drive solutions in developing software products. However, one of the interviewees was unable to participate in the online survey for whatever reason. This insight revealed that software developers often network with colleagues to discuss the business domain and technical problems over face-to-face meetings (i.e., offline communication), while also engaging in instant chat or messenger and Skype conferencing (i.e., online communication) to manage software tasks. The qualitative data for interviews were recorded, transcribed verbatim, and stored into NVivo software for analysis (Creswell, 2007; Saunders et al., 2009). Each interview transcript was labelled by pseudonym (e.g., AM) to represent the name of each interviewee. Based on the social network application and interpretation of the software development practices, pertinent keywords (e.g., sentences, phrases, or statements) were identified and coded into subcategories as open coding within each case of the interview dataset that reflected the underlying social network meaningful interactions and information needed for network insights. To find patterns of relationship for cross-case analysis, these subcategories of codes were further compared together in order to find common themes as the main categories based on axial coding criteria. This follows a similar approach in a recent study (Salo, Makkonen, & Hekkala, 2020). For example, the main categories were project activities, problem approach, communication channels, knowledge access and project performance goals. Each of these themes and their integral relationships provides information about the state of social interactions of software developers. In terms of the nature of work roles, how problems get solved, where knowledge is sourced to perform what activity, and how such development effort is measured (see Table 8 & Figure 8 below). This backdrop helps inform the design of the sociometric network survey in phase 2.

Table 8. Coding Scheme, description, and data examples

Category	**Description**	**Data Examples**
Project activity		
Understanding user requirements	Software developer's view of their work role	*"I take one of those user stories and I do the coding, I do the testing and then I demo it to… my team and my project owner."*
Automating project applications	Software developer's effort toward web services	*"I'm currently focused on project applications, which is sort of development and operational problems and trying to automate the development of applications into the web."*
Maintaining software infrastructure	Software developer's effort toward maintenance services	*"My responsibilities are making sure that infrastructure is up to date and evolving, performant and that any new features in that area are kind of my responsibility as well to just disarm and design."*
Fixing software problems	Software developer's effort toward mitigating production issues.	*"I keep an eye on production so if there's a problem, our software eventually will go into production and be used by the customers, so we have to watch what errors come up and deal with them."*
Problem-solving approach		
Seeking help from a senior colleague	Software developer's effort toward resolving development issues	*"If I'm banging my head against the wall, which does happen, I would ask my scrum master."*
Exploring solutions and seeking validation from colleagues	Software developer's effort toward generating solutions to problems	*"It's we have a conversation, and we say, 'I had time to look at this and I found A, B and C, what do you guys think?'* *"I would ask if the approach I'm taking is the right one. If I'm asking someone with experience."* *"I think talking to others gives me multiple perspectives and I think all our work is some sort of problem-solving*

		for the most part and all problems resolved to be a census of ideas…it's a different way of looking at solutions as well."
Knowing knowledgeable colleagues	Software developer's effort in identifying potential problem-solver	*"I know my team now and I know who knows what in the team so if I have one problem I might go to 'x' while if I have another problem, I might go to 'y'. The team is diverse, and they have different abilities and different knowledge."* *"For the domain knowledge, from experience Software Engineers would know a lot of it, because they're used to working in the domain."*
Problem resolution: Tracking the root cause of problems	Software developer's effort in identifying the cause of development issues	*"But then in my job, there's a lot of troubleshooting and a lot of that is experience and kind of going well this is the problem and learning them well if we do this, we can exclude that's causing the issue and if we do this, we know this isn't causing the issue. So just kind of breaking it down and trying to work it out and that I think is just something that you gain with experience and by watching other software developers and seeing how they kind of work it out, you know."* *"If there are any issues, we troubleshoot those issues. We may raise a defect, which is, a defect in the method we use for making corrections to the system. So the defect will detail what the problem is and like a user story that will go in the backlog and you take defects off the backlog, implement the change, implement the test for that change and follow it through as it's deployed to the different testing environments to ensure that it is actually passing the test."*
Communication channels		
Preference for offline knowledge sharing with colleagues	Software developer's effort in communicating software knowledge with others	*"I prefer the physical, my colleague that sits behind me knows lots. I can turn my chair and ask him!"* *"I try to talk face to face with them because it tends to work better."* *"It's definitely like, a few people in a room with a whiteboard, that's all I'd want. I would prefer that to be online."*

		"For domain knowledge, it's more got to do with face-to-face interaction."
Preference for online knowledge sharing with colleagues	Software developer's effort in communicating software knowledge with others	*"The internet yeah or over – over messenger usually we talk, or we can have Skype calls."* *"I might email, sometimes I'll Skype."* *"I find this quite useful, yesterday or the day before we had a big issue, and we had to use Skype to talk back and forth with someone in Germany to sort things out."*
Knowledge access (knowing who knows what knowledge)		
Domain (business) knowledge	Software developer's effort in sourcing for knowledge	*"For the domain knowledge, from experience Software Engineers would know a lot of it, because they're used to working in the domain."*
Technical knowledge	Software developer's effort in sourcing for knowledge	*"Yeah, bottlenecks…technical problems, you know, things that we can't overcome ourselves. And we have a scrum master…and it is their job to fix those bottlenecks so that we can work."* *"To be a Software Developer I need technical knowledge, but if I don't have domain knowledge, I can't use my technical competence."*
The usefulness of knowledge shared	Software developers' perception of knowledge received.	*"I just get better and more proficient in what I do and be able to achieve the goals with better solutions."*
Project performance goals		
Applying knowledge to tasks	Software developers' perception of knowledge received.	*"I just get better and more proficient in what I do and be able to achieve the goals with better solutions."*
Achieving software quality	Software developer's effort in developing good software	*"A failed test would indicate that there's a problem. Defects being raised would indicate poor quality."* *"I suppose reliability of what we've produced if you know what I mean. So, I suppose fewer bugs."*
Ensuring speed of navigation of features	Software developer's effort toward designing a more efficient software	*"I guess for the management it will be the time that it's taking you to do something and the time that it takes navigation from the source to the end-user."* *"A very efficient software that doesn't meet your requirement is no good. Whereas if you have software that meets requirements, but has poor quality and*

		efficiency, it's not very usable, but it has a bit of value. But if you have software that doesn't meet requirements, it has no value."

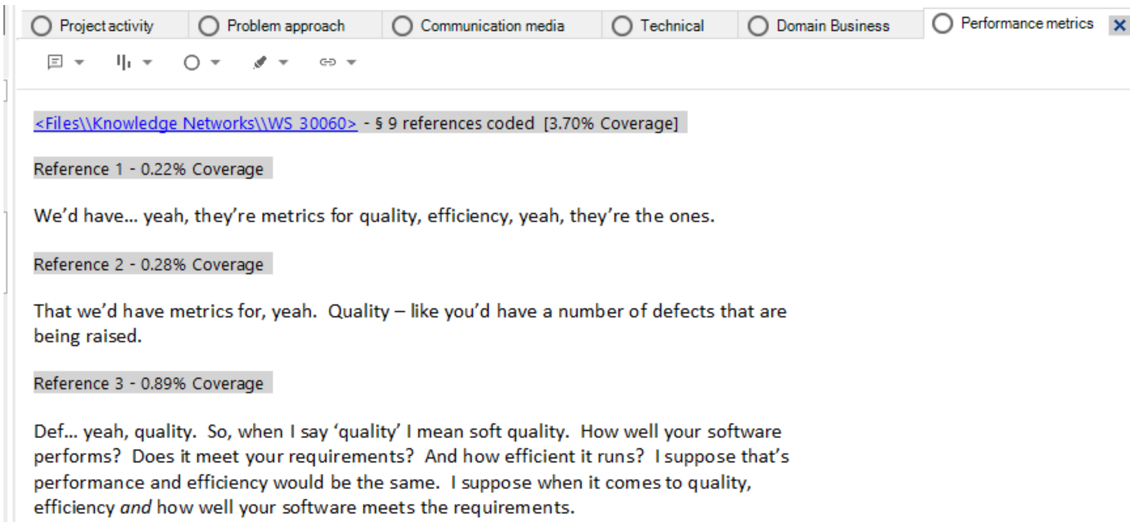


Figure 8. Understanding the social network of software developers through coding

In addition, Table 9 details those interviewees that have participated and can be identified in the network survey. By identifying these interviewees in the network survey conducted in phase 2, it shines a light on their brokerage roles and level of importance in the overall network with other team members.

3.5.2.2 Phase 2 – Network Surveys

Secondly, the insight generated from the interviews was used to design a social network survey, following the network survey tradition (Aalbers & Dolfsma, 2015; Whelan et al., 2011). This was administered to 35 software developers of which 30 people willingly participated. Approximately 25 people fully completed the online survey with a 69% response rate after discarding the incomplete surveys. Below provides a snapshot of the data collection phases and interviewee details, respectively.

Table 9. Interviewee Details

Node	Brokerage Role	Formal Position	Experience in Average Years
LD	Coordinator	Engineer	15
PS	Coordinator, Representative	Senior Engineer	17
PH	Coordinator	Senior Engineer	30
JM	Coordinator, Gatekeeper, Representative	Engineer	40
AM	Coordinator, Gatekeeper, Representative, Consultant	Engineer	11
ES*		Senior Engineer	15

* *The node ES is not on the network map due to non-participation in the survey.*

3.5.3 Measures and Analyses

The goal of this study was to understand how brokering activities in the business domain and technical knowledge sharing ties in offline and online networks contribute to software development. To capture such brokering interactions in software development networks, this study adopted a roster name generator methodology, which is widely recognized in social network research (Sykes et al., 2014; Zhang & Venkatesh, 2013). Roster name generator involves presenting a staff list of employees within a division and asking members with whom they share (and have shared) knowledge, including indicating the medium of interactions in this context. It is a retrospective self-reports approach to measure network questions. The same is used to generate offline and online brokerage in the business domain and technical knowledge network. Respondents were asked to "*Please identify which work colleagues you discuss technical problems with at least once a week?* The same question was repeated but replaced the word "*technical*" with "business domain". "*What medium do you usually use to discuss business problems with this person (select both options if applicable)*? Options were "*Offline*" and "*Online*", and whether "*Within my team*" and "*Outside my team*." However, the technical problems focus on understanding issues surrounding the use of technologies to develop and deploy software systems, and business problems as issues around understanding requirements, development processes, and workflow practices within their work or domain environment. These network questions were adapted from extant network studies (Belso-Martínez et al., 2017; Sykes et al., 2014; Zhang & Venkatesh, 2013).

Brokerage scores based on betweenness centrality by Freeman (1978) and honest broker metrics by Gould & Fernandez (1989) were computed from the UCINET 6.0 software for social network analysis by Borgatti, Everett, & Freeman (2002). While brokerage scores provided information about the key players or brokers in the networks, broker metrics provided more information about the number of times an individual occupied one of the different brokerage roles in these networks. To note, higher betweenness scores or higher betweenness centrality has been linked to the role of an innovation broker in gatekeeping capacity, and an increase in information and code quality in producing software (Hossain & Zhu, 2009; Sasidharan et al., 2012; Winch & Courtney, 2007). Together, this generated better insights into the brokerage process of software developers. This software package was used to capture the x-ray visualization of the network positions of brokers among a set of software developers embedding in the whole network of business and technical knowledge networks, and their local structures in the offline and online business domain and technical knowledge networks, the main point of reference, comparatively.

3.6 Results

3.6.1 Insight from Interviews: Understanding software development knowledge sharing

Based on the software-oriented organisation, the interviews revealed that software developers' roles involved processing "user stories" for operational use by end-users. User stories (old language was called requirement specifications) within Agile were described as some requests made by someone asking for an increase in functionality of a feature to be developed as an added value to customers.

On the role of knowledge resources needed in developing such a software product, the interviewees mentioned that business domain knowledge and technical knowledge were required for its development. The business domain knowledge was described as real-world knowledge used in understanding the problem surrounding a given product, gaining an awareness of the business mindset and plan around the product, and the new methodologies and technologies to develop and make such a product work or behave correctly. Whereas the technical knowledge was said to be relevant in understanding how technologies and the architecture interact together (software systems and applications), which included writing and delivering the codes needed to drive the right solution to the given problem.

On the importance of each type of knowledge, an interviewee LD disclosed that the absence of one of the knowledge resources rendered the other one inactive, or unusable. Meaning that his technical know-how of programming languages and technological tools would need to be complemented with business domain knowledge:

> *"To be a Software Developer I need technical knowledge, but if I don't have domain knowledge, I can't use my technical competence."*

They explained further that the only way to get business domain knowledge was by acquiring and reading relevant documents and engaging in personal conversations around the domain area, which often required face-to-face conversations. Interestingly, this study found that business (domain) knowledge was often sourced from experienced systems managers and that a new software developer in the company might lack this type of knowledge.

Identifying these two knowledge sources is important to bridge the value and knowledge gap between what customer needs are, and what the organisation can equally deliver when determining requirements in software development. This is pivotal in understanding the nature of knowledge to be acquired, negotiated, and integrated to solve development problems, drive solutions to users, and achieve successful software development (Coughlan, Lycett, & Macredie, 2003; Mehta & Bharadwaj, 2015). So, the competent know-how in matching customer needs with the right technologies from the operational perspective would be expected to deliver customer solutions, and ultimately software quality. Therefore, the inability to access this complimentary knowledge when needed can delay software development effort on the productivity of team members to achieve individuals and collective project goals (cf. Cataldo & Herbsleb, 2008).

Given the uncertain nature of the development process, changes in requirements might deserve new methodological and technological solution-driven approaches that will motivate software developers to avail of their embeddedness in any of these knowledge networks to solicit broker's support. Without understanding the business and technology side in parallel, software developers might find it difficult to use the right technology to deliver the best solution or be unable to meet requirements, which may result in poor quality of software or producing efficient software that may be unusable.

More importantly, by identifying this type of knowledge requirements (business/IT domains), it gives insight as to why software developers might be compelled to be embedded or operate in different knowledge brokering activities for much-needed knowledge resources in software development (Whelan, Anderson, van den Hooff, &

Donnellan, 2015). This can also partly influence how each software developer performs in handling complex problems and coordinating software development tasks in general. Effective communication of these knowledge types is thus important in determining and developing software requirements (Cataldo & Herbsleb, 2008; Coughlan et al., 2003; Sawyer et al., 2010).

3.6.2 Insight from Social Network Analysis: Understanding individual brokerage dynamics

SNA was applied to identify brokers to see if the same brokers operate in business and technical knowledge sharing and across offline and online communication networks. Software developers are denoted by different node pseudonyms (alias, for example, AG, CC) on the network map. The bottom line is that software developers make use of business domain knowledge and technical knowledge they receive from and provide to other co-developers to solve and guide solutions to problems to drive performance individually and collectively. The outcome of this analysis showed that brokers (core software developers) and brokerage roles (coordinating, gatekeeping, etc.) vary in different knowledge (business/technical related) and communication structures (offline and online networks).

Table 10. Brokerage scores based on betweenness centrality (Freeman, 1978)

		Business Network			Technical Network			
	Node	Outgoing Ties	Incoming Ties	Betweenness Score	Outgoing Ties	Incoming Ties	Betweenness Score	Hierarchy
1	JC	2	0	0	5	3	83.03	Senior Engineer
2	AG	3	4	27	3	4	169.03	Senior Engineer
3	AD	2	2	0	6	5	212.32	Senior Engineer
4	CC	2	6	48	5	3	89.42	Senior Engineer
5	PS	1	2	3	3	3	94.2	Senior Engineer
6	NB	0	0	0	1	2	0	Engineer
7	DO	0	0	0	2	4	0	Senior Engineer
8	FC	1	0	0	4	4	14	Engineer
9	SG	0	0	0	3	3	0	Engineer
10	AM	4	3	32	5	3	5.42	Engineer
11	KW	0	1	0	2	2	22.75	Engineer
12	HG	6	1	10	6	3	47.12	Engineer
13	BM	1	0	0	4	3	4.92	Engineer
14	MF	1	1	0	1	2	19.83	Engineer
15	CE	1	1	0	2	5	14.17	Engineer
16	LD	1	1	0	2	3	27.8	Engineer
17	PH	0	1	0	1	3	23	Senior Engineer
18	AM	2	0	0	6	1	132.27	Senior Engineer
19	TJ	4	2	0	4	3	4.33	Engineer
20	JM	2	2	0	5	4	39.5	Engineer
21	EM	0	4	0	3	4	34.62	Senior Engineer
22	AO	2	2	0	0	5	0	Engineer

23	MT	2	3	2	4	5	21.43	Engineer
24	PO	2	2	0	5	5	216.37	Engineer
25	A0	2	3	3	5	5	49.43	Senior Engineer

The findings in terms of brokerage profiles identified are summarised in the following table, outlining the main contrasts identified in the various brokerage roles encountered:

Table 11. Number of Times A Software Developer Takes on One of Four Brokerage Roles

Communication Network	**Coordinator (i)**	**Gatekeeper (i)**	**Representative (e)**	**Consultant (e)**
Business domain knowledge network	13	8	6	17
Offline business domain knowledge network	13	12	10	7
Online business domain knowledge network	--	--	--	--
Technical knowledge network	74	9	12	16
Offline technical knowledge network	51	3	12	21
Online technical knowledge network	17	--	5	--
Brokerage* orientation i = Internal Brokerage orientation e = External *Notes*: total number of software developers in each role, in each network – software developers can potentially combine different roles in the same communication network. Brokerage orientation: internally oriented role (coordinator, gatekeeper) and externally oriented role (representative, consultant).				

*Brokerage scores based on Honest Brokers (Gould & Fernandez 1989)

Table 11 presents different brokerage roles in the offline-online business domain knowledge network versus offline-online technical knowledge network. To reiterate, a coordinator connects two members of their group with knowledge. A gatekeeper obtains knowledge from outside and then shares it with members of their group. A representative retrieves knowledge from their group and then shares it with members of another group. A consultant shares knowledge with two members of the same group. Thus, the above table shows that project members of different groups can combine communication roles within and between groups to act in similar and different knowledge brokering in the offline versus online business domain and technical knowledge networks.

Surprisingly, it appears that no identifiable communication roles in the online business domain knowledge network, including the lack of the roles of gatekeeper and consultant in the online technical knowledge network. Whereas a software developer might for instance be playing the role of a gatekeeper by externally scouting and filtering for relevant resources of either or both business domain and technical knowledge related that may be lacking and can further add to the knowledge of their groups. While also being a coordinator by joining disconnected individuals within their group with any of these resources, such as for stimulating creative interactions internally. Combining these roles can be valuable in leading software innovativeness or innovation development in software projects (Aalbers & Dolfsma, 2015; Jordan & Segelod, 2006; Whelan et al., 2011). Likewise, individuals playing the role of representatives and consultants can decide to share business and technical knowledge resources beyond their group to bridge disconnected members that can foster the innovativeness of another group. In other words, a situation where a broker may inspire external members to be more creative in approaching software development problems, or rather become more innovative through an inter-team network bridging, without being officially mandated to do so, or forced to hold dual membership in another team project. Recent research has shown the effect of collaborative inter-team ties on greater network performance (Wu et al., 2021).

This study found that software developers can act as coordinators who might likely be leaders of their group to connect them with knowledge resources to perform software tasks. More so, individuals performing the roles of coordinators and consultants tend to serve as brokers of the business domain and technical knowledge resources between internal and external group members. However, this table does not show the role of liaison or itinerant, which would have equally benefited bridging between internal and external team members in software development knowledge sharing. The specific dynamics of brokers are provided below.

3.6.2.1 Business Domain Knowledge Brokering

In the parent network excluding the online versus offline brokerage, the brokers (core software developers) were AG, AD, and AO. Compared to the actors that occupied a less dominant position (for example see CC) and other peripherals (for example see PH, SG). Actor AG and AO (alongside CC) appeared to have a higher number of direct and indirect personal ties to less connected others and were therefore positioned as links within this network to provide knowledge and informational benefits to their immediate contacts, in addition to getting these resources disseminated and utilised by their surrounding neighbours.

Furthermore, their social attributes revealed that they held senior positions in their work roles as project leaders in the organisation. In other words, these core software developers would be described as more knowledgeable and tended to be accessing, controlling, and providing business domain knowledge inputs to others in their networks. This can be associated with their professional years of expertise and the length of time being functional in this organisation. In this study, it appeared that each of them was acting more or less in gatekeeping, consulting (and supervising), representing, and coordinating functional roles to manage and facilitate access to business domain knowledge resources within and outside their groups. As many people sought them for business-related resources to solve their problems and supposedly influence performance.

However, their removal from the network can also render this network fragmented to stall knowledge resources flow for coordinating work. The population of the software developers in this network appeared to be more homogenous. This suggests that that the network was more centralised around a few software developers dictating access and control of vital business domain knowledge and informational resources.

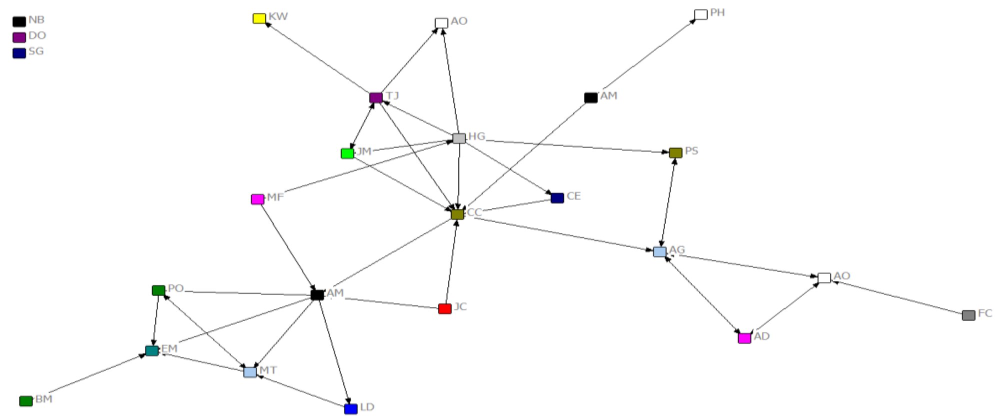


Figure 9. Business Domain Knowledge Network

At the local network level of viewing business domain knowledge brokering in the offline and online brokerage, software developers in the offline brokerage were CC, AM, and seemingly non-existent in the online brokerage. However, the lack of ties let alone brokers in the online brokering of business domain knowledge is a striking finding. Yet one of the previous interviewees had suggested that acquiring business domain knowledge was largely a face-to-face conversation that was not popular virtually. Besides, the role of CC would suggest acting as an intermediary between unconnected direct contacts to channel business domain resource benefits towards them. Hence, CC would be perceived as performing gatekeeping and consulting roles in the network. However, actor AM also appeared to be receiving and sending business domain-related resources in and out of their cohorts, thus seemingly acting in gatekeeping and representing capacity.

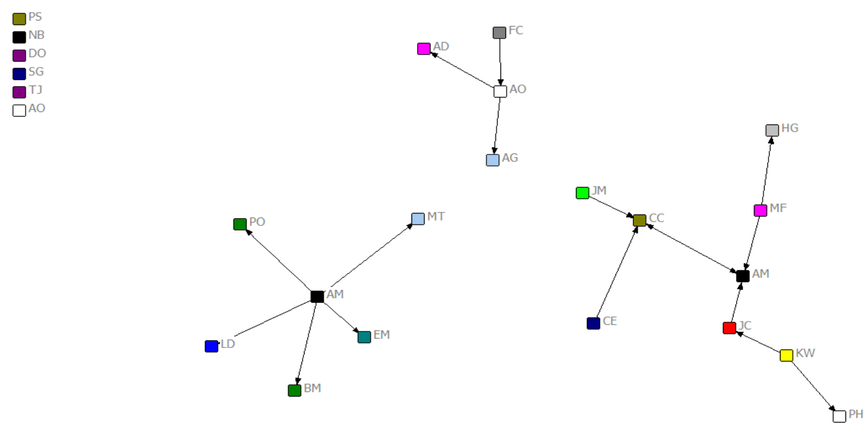


Figure 10. Offline Business Domain Knowledge Network

3.6.2.2 Technical Knowledge Brokering

Unlike the previous approach, the whole network configuration of technical knowledge was considered before analysing the local structures of offline and online brokerage. An understanding of the whole network and local structures is important to compare and analyse changes and differences in varying brokerage roles because individuals might be connected to play diverse roles or not in different sub-network structures among software developers. Hence, brokers engaging in brokering technical knowledge in the whole network were AO, AD, DO, FC, and SG. A few less core software developers (for example see CC, PS) were seemed to be connected to these main players, and such connections would be needed for communicating relevant technical knowledge resources, and for coordinating and performing tasks by others. In general, AD seemed to be playing more coordinating roles, while AO played more consulting roles. The population of the software developers appeared less homogenous but instead heterogenous. This would suggest that the network was less centralised or concentrated among a few members who were able to provide access to relevant technical knowledge and informational resources.

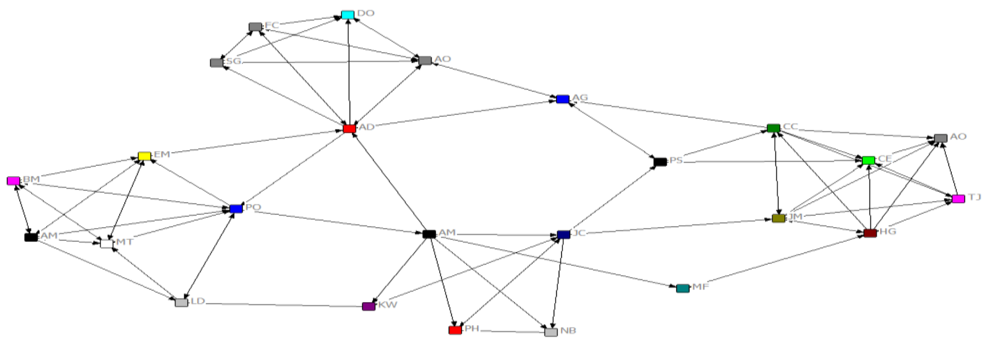


Figure 11. Technical Knowledge Network

Considering the local structures in technical knowledge brokering, the offline and online brokerage showed the evolving and varying nature of brokers. Specifically, in the offline brokerage of technical knowledge, the brokers were actor AD, PS, and KW. While actor AD was more active in coordinating relevant technical resources to co-developers than actor PS, who seemed to be acting in a minimal representative role. In contrast, the online technical brokers were actor BM, EM, PO, and AM. With the exemption of actor AM who would be considered as more active in consulting roles, the rest were into coordinating and representing roles to a lesser degree.

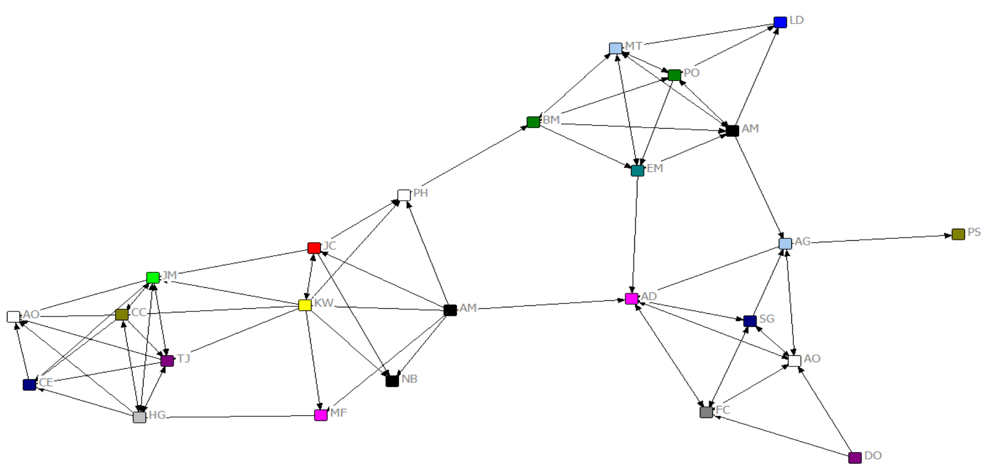


Figure 12. Offline Technical Knowledge Network

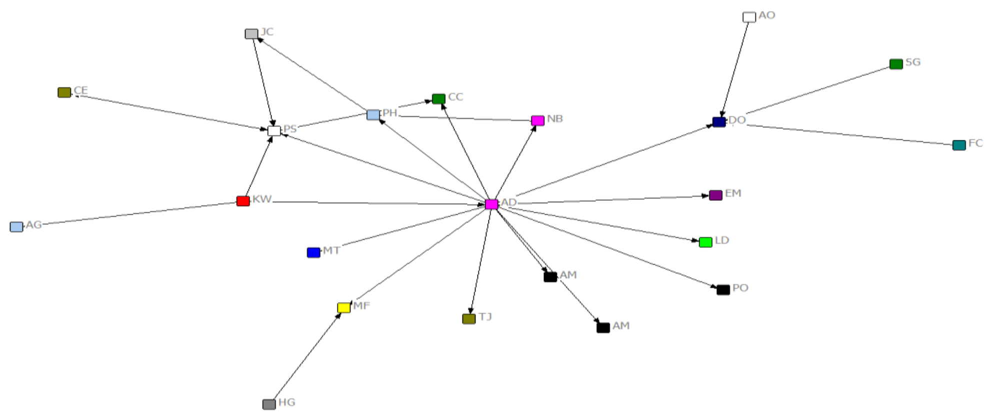


Figure 13. Online Technical Knowledge Network

3.7 Discussion and Conclusions

The purpose of this study was to understand the dynamic roles of brokers in offline and online brokerage in the business domain and technical knowledge sharing in software development. This study shows that software developers are largely driven by informal interpersonal knowledge brokering activity among project members in facilitating software development knowledge-sharing efforts. Thus, this study identifies and discuss the following in detail.

3.7.1 Theoretical implications

There is empirical evidence that has shown the importance of creating alignment between business and IT domains or in integrating both business domain knowledge and technical knowledge between business and IT units (Cataldo & Herbsleb, 2008; Mehta & Bharadwaj, 2015; Ryan & O'Connor, 2013; Šmite et al., 2017; Tiwana, 2004, 2012; Vermerris et al., 2014; Whelan, Anderson, van den Hooff, et al., 2015), even beyond software development in organisations (Aalbers & Dolfsma, 2015; Belso-Martínez et al., 2017; Reus et al., 2020; Whelan et al., 2010); yet, understanding the dynamics of brokers in the diverse business domain and technical knowledge sharing in offline and online brokerage combined remain largely ignored, or underexplored.

In bridging this gap, the result of the face-to-face offline brokerage revealed that individuals engaged in combining and performing different brokerage roles: coordinator, gatekeeper, representative, and consultant in both the technical and business domain knowledge sharing. In the online brokerage, while individuals performed the role of coordinator and representative, the role of gatekeeper, consultant and liaison were missing in technical knowledge sharing. But there was no knowledge sharing activity whatsoever in knowledge brokering capacity in the business domain knowledge network. This result is surprising yet interesting because the inability to perform these missing roles can affect software developers' ability to seek additional business domain knowledge resources outside the group that might be lacking (in addition to technical knowledge) and to absorb and integrate them for use inside the group. In other words, the inability to build on their business domain knowledge including the team's knowledge base that may be necessitated in dealing with requirements' changes, can impact achieving software development success vis-à-vis resolving software problems quicker, and driving software innovativeness more effectively.

Offline knowledge brokerage

Our important contributions extend the findings of a number of recent empirical studies. First, this study goes beyond drawing inferences qualitatively in offline brokerage on the importance of scouting and bridging in the business domain and technical knowledge sharing among software development teams (Šmite et al., 2017). Also, Mehta & Bharadwaj (2015) have shown a positive interaction of the performing role of sentry (gatekeeper) and guard (representative) activities relating to business domain and technical knowledge sharing on software development team performance. This research showed that performing the role of the gatekeeper at a high level is more critical to reducing the risk of high uncertainty, thus fostering greater knowledge integration than performing guard activity under high uncertainty, which tends to reduce knowledge integration during the software development process. However, such contributions are limited to the role of gatekeeper and representative and offline brokerage. Similar studies that have examined similar brokerage roles did so by examining them across the stages (determining requirements/development) of software development within the offline brokerage. But also fail to consider their dynamism in the business domain and technical knowledge relations, and in online brokerage despite drawing their relationship on different performance consequences in software development (Sawyer et al., 2010). A similar limitation was also observed in non-software development settings, where some of the brokerage roles were positively associated with organisational innovativeness (Aalbers & Dolfsma, 2015). Interestingly, where brokerage in business domain knowledge and technical knowledge sharing were related to innovation performance, it was however limited to structural hole measure and offline brokerage (Belso-Martínez et al., 2017). One of the contributions is extending the general role of boundary spanner in knowledge sharing observed in offline brokerage (Boden & Avram, 2009).

Online knowledge brokerage

Second, this study contributes to understanding the quality of knowledge brokering beyond the ability of brokers to expand the knowledge of their groups in online brokerage (Reus et al., 2020) by paying attention to the actual knowledge types (business domain/technical knowledge), and how it influenced dynamic brokerage roles, or network bridging in an online environment. Because the presence or absence of brokerage roles (e.g., gatekeeper/consultant) may affect the ability of a team member to generate external knowledge, let alone expand the transactive memory systems (knowledge) of their team. An existing study has linked how the variation in team's knowledge or transactive memory systems differentiated low and high-performing software development teams (Ryan & O'Connor, 2013). Including extending other offline studies based on the relation between brokers and the exact type of information and knowledge usefulness in network brokerage dynamics (Aalbers & Dolfsma, 2015; Whelan et al., 2010). This shows how the diversity of knowledge and communication networks can alter the dynamics of brokers. In this offline brokerage, Cataldo & Herbsleb's (2008) study revealed the important role of gatekeepers in navigating technical and business domain expertise, and hence showed their positive effect on the software development effort. Observing more brokerage roles and beyond online brokerage also extend our contributions of brokerage roles, knowledge, and communication networks.

Evolving nature of brokers

Third, the result revealed that different brokers (not just brokerage roles) emerged in offline and online brokerage in the business domain and technical knowledge sharing. It is assumed that the homogeneity of business domain knowledge network compared to the heterogeneity of technical knowledge network might have played a role in more concentration of brokerage roles and varying brokers emerging in technical than business domain knowledge brokering activity. This may depend on whether individuals utilise their contacts to source and share business domain knowledge and or technical knowledge to perform tasks, and whether any of these knowledge types can be accessed either through an offline brokerage or online brokerage, or both. At the organisational level, scholars have attributed status to an actor’s affiliation in a business domain knowledge network and proximity in a technical knowledge network, whereas embeddedness tends to drive affiliations in both knowledge networks (Balland et al., 2016). More importantly, the findings revealed that some brokers appeared to be brokering for both knowledge resources, and across offline and online brokerage in parallel (multiplex brokering). This partly extends studies seeking more clarity concerning why individuals may be embedded in diverse resource networks simultaneously (Kwon et al., 2020; Podolny & Baron, 1997).

Lastly, it is uncovered that the formal positions that brokers officially hold in this organisation tend to influence their network positions and roles in diversely accessing and sharing business domain knowledge and technical knowledge across their contacts in this offline and online brokerage. Because many of the brokers in business domain knowledge and technical knowledge sharing appeared to be high ranking members. For example, being a scrum master and technical lead architect seemed to have shaped their central positioning because they were more connected in diverse knowledge interactions. Thus, acknowledging the functional influence of employees' attributes on their network positions within their knowledge networks in software development knowledge-sharing efforts among software developers. This finding also supports the findings that showed technical leaders being more instrumental in information sharing, and in increasing the productivity of the software development teams (Ehrlich & Cataldo, 2014; Šmite et al., 2017). Similar network studies have shown that the interactions between structure, network position, and individual attributes tend to affect access to more support resources to outperform others (Bilecen & Cardona, 2018).

3.7.2 Managerial implications

It is well established that software development effort requires a high level of problem-solving, creativity, and knowledge sharing to drive its success in organisations. Realising this core determinant has influenced scholars to seriously pay attention to its development (Dissanayake et al., 2013; Gallivan, 2003; Mishra et al., 2012; Payne et al., 2018; Wang et al., 2012). The focus on software development networks on the role of knowledge brokering in offline and online communication networks requires special attention to exploring and maximising the human side of software projects. This study explored four brokerage roles: coordinator, gatekeeper (both as internal oriented), representative, and consultant (both as external oriented) within and between offline and online brokerage in business domain knowledge and technical knowledge sharing among software developers. The finding revealed that individuals who are predominantly internal and external oriented in their offline contribute to software development knowledge sharing activity within the organisation. Also, individuals who in their online (but not in their offline) contacts are internal and external oriented will particularly contribute to software development knowledge sharing, potentially in driving greater problem solving, designing software, and delivering software quality within the organisation. These individuals would be more collaborative, and not be afraid to network across other teams.

Recognising such individuals in software development knowledge sharing would be beneficial for both intra-and inter-team knowledge networking and organisational-wide collaboration at the informal and formal levels of engagement. Hence, organisational managers are encouraged to pay more attention to them.

Furthermore, in online brokerage individuals were not utilising this avenue to create network ties, much less occupy separate and different brokerage roles in sharing business domain knowledge resources with others (cf. Whelan et al., 2015). Considering the current COVID-19 pandemic where software development teams are now remote workers (O Connor et al., 2021), how organisational leaders cultivate online brokers to effectively share business domain (beyond technical) knowledge resources to co-developers will be crucial for effective brokering and collaboration in digital space among software development teams. This may be warranted to officially assign online brokers (more knowledgeable and experienced across business domain and technical domains) – in addition to the more effective emerging ones – and position them at the centre where they would be very well connected, easily reachable, and available to render functional support to peripheral specialist others. Exploring both formal and informal networks, in general, has equally been of interest to scholars (for example see Cataldo & Herbsleb, 2008; Šmite et al., 2017). This would be to bridge the knowledge gap that may have existed, or is bound to exist, in timely accessing and integrating this diverse knowledge to solving complex and tricky problems, plus increasing software development knowledge-sharing effort, and eventually shaping productive and task coordination success. Especially for new but not tenured software developers in the organisation who might find it difficult to easily access business domain knowledge experts to complement their technical competencies in determining requirements and performing development tasks (Ryan & O'Connor, 2013). Overall, the use of technologies (e.g., email, intranet, discussion forum) in online brokerage can certainly play a role because they provide a broader reach to facilitate, disseminate and utilise knowledge resources, be it technical and business domain related.

Recognising the different brokerage roles and those occupying them can help managers and leaders on how to encourage their effective functioning in knowledge sharing effort in communicating, collaborating, and coordinating tasks. This is because some brokerage roles and individuals might be good at generating new ideas, others in connecting and developing them, while some others can potentially combine them. Insofar, an understanding of the strengths and weaknesses of each network brokerage (offline, online; coordinator, gatekeeper, etc.) can help capitalise on their combinations and attractiveness in software development knowledge-sharing efforts.

Furthermore, an understanding of the dynamics of brokerage roles by organisations and project managers can be driven when there is a support and reward system to encourage organisational-wide knowledge assistance, where those who contribute to both intra-and inter-team problem-solving, creativity, and innovativeness are rewarded (e.g., in form of tangible gifts, bonuses, pay raises, and award recognition) by their knowledge brokering effort. Why is that the lack of adequate support in combining multiple roles in knowledge brokering activity may be stressful, especially when such individuals are acting in unofficial capacity based on their colleagues' recommendation, or where interpersonal feedback culture is encouraged by the management. This study suggests that organisational leaders and managers pay close attention to cultivating online (and beyond offline) brokers as society widely embraces the reality of online social interactions in the COVID-19 disruption and digitisation era. All these can improve the success of software development projects.

3.7.3 Conclusions, Limitations, and Future Research

In this study, the goal was to understand how the nature of brokerage in different knowledge interactions and communication networks combined influenced software development knowledge sharing and the success of software projects. In doing so, this study addressed the gap in the literature on knowledge brokering processes in the offline and online brokerage relating to business domain knowledge and technical knowledge sharing towards knowledge distribution (Boden & Avram, 2009), knowledge expansion (Reus et al., 2020) and knowledge integration (Mehta & Bharadwaj, 2015; Šmite et al., 2017) in software development settings.

By comparing business domain and technical knowledge sharing in the offline and online brokerage, it is found that the patterns of brokers and communication roles are dynamic when viewed from different knowledge and communication networks, which altogether advanced an understanding of software development knowledge-sharing effort among the observed software developers. More precisely, the knowledge brokering activity does not always follow the same patterns within and between different knowledge and communication networks, and the level of involvement for example in the role of coordinator and consultant, or gatekeeper and representative activities can optimally, or sub-optimally affect the ability to distribute, expand and integrate knowledge differently among software development teams, and on other performance-related outcomes (e.g., problem-solving, innovativeness, software quality, or task effectiveness).

Viewing the role of brokers as dynamic and not necessarily limited by project membership can help every team member to contribute more meaningfully to task effectiveness, and overall fulfilment of the individual, project, and organisational goals. The interaction of software developers' characteristics in terms of who they are (e.g., official positions) and contributions (e.g., brokerage positions) in software development knowledge sharing does help to see the impact of diversity through dynamic participation across communication networks. However, an environment that favours Intra – and inter-team connectivity for creative, innovative, and collaborative knowledge interactions would provide a fertile ground for learning, support, and greater performance of individuals and teams.

As with most studies, these contributions are limited. Due to the small number of software developers surveyed in this study, it is difficult to ascertain the actual composition of software development teams and who the actual project leaders are, apart from those observed as project leaders in this organisation. But then, the inability to see the same individuals emerge across business domain knowledge and technical knowledge in offline and online brokerage make interesting the dynamic nature of brokers. This study could not compare if any of the observed brokerage roles and their high or low involvement may be insignificant when observed across different stages of the software development process, and on performance differentials, as observed by Sawyer et al. (2010). Therefore, longitudinal rather than cross-sectional studies can help in understanding whether any of these differentiated brokerage roles might change or remain across the stages of the software development process.

Therefore, understanding how variances of knowledge brokering activity in the business domain and technical knowledge sharing and communication networks affect key performance outcomes would warrant access to some relevant performance measures in place for evaluation. Such variances, for example, can be related to network relationships involving purely business domain or technical knowledge sharing, and their multiplex interactions on, for example, an individual problem-solving competence – based on the ability to solve problems and complete tasks on time; software quality–based on the number of defect counts in a team; collaboration – based on the willingness to help others on arising problems while coordinating work; software innovativeness – based on the creative ability of team members to come up with software designs of features.

All these would depend on the overall project performance goals and available metrics set out by the organisation to evaluate project teams and are willing to be shared for social network studies. Given that not all the observed software developers are well connected within and across these knowledge and communication networks but tend to be siloed. This also deserves how best to structure tasks to promote effective offline and online communication of diverse knowledge resources among project members by brokers. More especially, in officially cultivating online brokers among software development teams in an online environment. This study is limited to a single software-oriented organisation and hence its findings cannot be generalised. Yet it presents areas of interesting considerations by organisational leaders and further research endeavours for academicians in similar work environments.

Chapter 4 The Effect of Advice Network Structure on Problem-Solving Competence Among Software Developers

4.1 Introduction and Chapter Outline

This chapter provides an intersectional study of previous chapters. It examines the advice structure and brokerage behaviour of software developers in multiplex networks and the way such interactions with others dictate an individual problem-solving competence in the software development effort. Section 4.2 relates to the abstract of this study. Section 4.3 introduces the research context. Section 4.4 considers the theory and hypothesis development on the different constructs of this study. Section 4.5 examines the research method in gathering the relevant data. Section 4.6 focuses on analysing data and presenting the results. Section 4.7 provides a discussion of results and conclusion. Section 4.8 details the limitations of this study and future research directions. Lastly, section 4.9 wraps up with a conclusion.

4.2 Abstract

Software development requires software developers to share knowledge and solve problems together. Although researchers have considered the business and technical knowledge resources germane to performing software development tasks, empirical studies investigating both business and technical advice networks on problem-solving competence of software developers is scarce. Using social network theory, this study argues that software developers must be embedded for knowledge brokering within and across business and technical advice connectedness for improving problem-solving competence. But that such connectedness is moderated by contact quality towards developing the problem-solving competence to contribute efficiently and effectively to software development effort and success. The network data was collected through an online survey from 153 respondents in a software organisation. The findings suggest that a software developer who engages in knowledge brokering in business and technical advice (network) connectedness will particularly increase problem-solving competence

to drive their software development effort. However, and surprisingly, this result revealed that contact quality played no significant moderating role between these advice networks and the problem-solving competence of software developers. Based on these findings, this study discusses the implications for theory and practice.

***Keywords**:* advice network; business advice connectedness; technical advice connectedness; software development; problem-solving competence; software developers; contact quality; brokerage

4.3 Introduction

There is much empirical evidence relating to the effect of social network structures on different performance outcomes in software development (Bernardi et al., 2018; Peng et al., 2013; Sykes et al., 2014). For example, some of these studies have emphasised the role of social networks, specifically advice networks, in helping individuals search for knowledge required to fill a work-related knowledge gap (Brown & Duguid, 1998; McGrath, Vance, & Gray, 2003). The quality of contacts in social networks, however, has also been considered beneficial, in addressing work demands and improving socioemotional benefits among coworkers (Burmeister, Alterman, Fasbender, & Wang, 2021; Fasbender, Burmeister, & Wang, 2020). This includes the ability to create, generate, and share knowledge with others to solve problems, promote learning, and achieve task performance (Gibbons, 2004; Siciliano, 2015; Sykes et al., 2014). More importantly, studies have shown the role individual players or brokers' embeddedness in technical advice networks plays in facilitating knowledge sharing to improve software development effort and performance of software development teams (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014). Other studies have linked software developers' embeddedness in both the business domain (knowledge of general business processes and customer needs) and technical advice (technology knowledge needed to solve and deliver customer solutions) networks, for example on the team performance (Mehta & Bharadwaj, 2015; Sykes et al., 2014; Tiwana, 2004). In enterprise systems implementation and drawing from social network theory, Sykes et al. (2014), for instance, suggest that access to business advice (workflow), help fix non-software related organisational tasks while technical advice (software) help fix system errors and constraints related to technology tasks in performing software applications.

Social structures matter to problem-solving. Software development is a problem-solving activity that often demands software developers within information systems and software engineering to drive solutions to technical problems (e.g., fixing software bugs) and business domain problems (e.g., interpreting requirements) during the development process (Conaldi & Lomi, 2013; Rus et al., 2002; Tiwana, 2004). Solving these problems requires access to and integration of both business domain and technical advice to improve problem-solving skills (Sykes et al., 2014; Tiwana, 2004). Lin et al. (2015) for example find that access to both types of knowledge increase the team's problem-solving competence and project performance. Likewise, the reactive and anticipatory (proactive) mechanisms (e.g., available and findable expertise, experience and knowledge resources) in the organisation have also been linked to increasing the team problem-solving competence needed to address requirements uncertainty at different development stages in software development to maintain software quality (Yuzhu Li, Yang, Klein, & Chen, 2011; Sykes et al. 2014). Other studies have examined the alignment between the social and technical capability of software development teams on the software development performance (Maheshwari et al., 2012). Overall, studies have shown that individuals' interactions with a pool of information and knowledge resources and their collective integration drive timely, less costly and quality solutions to problems, and have resulted in efficient and cost-effective product performance (Sheremata, 2000).

Despite these contributions, limited empirical evidence exists regarding the roles of advice network structures and their impact on the problem-solving competence (i.e., developing the capacity to solve problems) of software development professionals (Lin et al., 2015; Siciliano, 2015). This is important because advice networks are important in explaining performance outcomes. To date, however, there is an incomplete picture of the causality undergirding this interaction, as prior studies often rely on a uniplex instead of a multiplex view of networks. Prior work, for example, resides on a conceptual design where business domain and technical advice interactions are studied separately in their respective software development settings (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014). Their study apart does not offer greater insights into the dynamics of these networks, which the multiplex network approach can address (Schreiber & Zylka, 2020; Tiwana, 2012; Vermerris et al., 2014). This study extends our understanding of the effect of advice network structure on problem-solving competence by incorporating a multiplex network approach to these advice structures. In doing so this study shed light on interactions and variations between diverse yet overlapping networks in a software development context.

A multiplex view, as opposed to a uniplex view, considers more than one advice network relation or observations between actors, hence, allowing for conceptual overlap in a single study (Belso-Martínez et al., 2017; Snijders et al., 2013). For example, a multiplex study by Belso-Martínez et al. (2017) found that structural brokerage (the role of actors in connecting disconnected others with knowledge resources) in business and technical advice networks results in improved innovation performance between organisations. Understanding the antecedents of problem-solving competence has been at the team level of analysis in prior software development research (e.g., Atuahene-Gima & Wei, 2011; Yuzhu Li et al., 2011; Lin et al., 2015), which makes applying a multiplex view to such team interactions. Insofar, the research question is: *What relationship between business and technical advice networks and their brokerage activities affect a software developer problem-solving competence?* To theoretically and empirically capture the dynamic of different knowledge interactions in informal advice networks in software development, below present theory, and hypotheses development.

4.4 Theory and Hypotheses Development

4.4.1 Problem-Solving Competence and Software Development

Problems are part of human activity, be it task-related or not, and are varied in different intensities. The extent to comprehending such problems can determine whether it can be self-managed or needs seeking interpersonal advice (MacGeorge, Feng, Guntzviller, & Feng, 2016). Because individuals differ in their cognitive load to process complex work-related problems (Burmeister et al., 2021). Problems, on the one hand, are described as knowledge gaps between the present state and an end state (Lin et al., 2015; Thomas, Lee, & Danis, 2002). Competence, on the other hand, refers to a specialised system of knowledge, abilities, cognitive and domain skills, proficiencies, and behavioural attitude of individuals and teams to be proactive and reactive to performing familiar and unfamiliar tasks (Amabile, 1983; Li et al., 2011; Lin et al., 2015; Šmite et al., 2017). Perry-Smith (2006) links cognitive and domain skills to creativity by being divergent, flexible, and evaluative in synthesising unrelated concepts and making sense of them in guiding solutions to problems (Aldave, Vara, Granada, & Marcos, 2019). This is why problem-solving through creative solutions goes beyond an established convention in synthesising processes, information, and knowledge, to charting a new solution path forward toward achieving performance (Amabile, 1983; Hoegl & Parboteeah, 2007).

Therefore problem-solving competence is concerned with the process of identifying, clarifying, evaluating, and reviewing alternatives, in implementing creative, and cost-saving solutions to a problem (Açıkgöz & Ö. İlhan, 2015; Atuahene-Gima & Wei, 2011; Lin et al., 2015; Sheremata, 2000). Primarily software development revolves around requirement analysis, design, development (coding/configuring), testing, and deployment activity (Iden & Bygstad, 2018; Sawyer et al., 2010). Hence, software developers are often charged to solve quality-related problems such as eliminating bugs and addressing design, comprehension, and redundancy problems (Balaban, Maraee, Sturm, & Jelnov, 2015; Coughlan et al., 2003; Pitts & Browne, 2007). However, research suggests that brokers are needed to facilitate socialising and communicating with others in order to share knowledge and improve shared understanding, which is often required to comprehend and solve work problems during software development (Boden & Avram, 2009). Researchers have also argued that recipients of knowledge can experience overloading and depletion of knowledge if they consult with low-quality contacts. But that such cognitive costs can be mitigated when the contact quality is high, as concrete and less ambiguous knowledge are shared by the providers to drive action and influence positive outcome (Burmeister et al., 2021). Therefore, the need to develop or increase problem-solving competence is critical in driving software development effort. Since individual competence requires cultivating the right knowledge, skills, attitude, and motivation to put them into action towards providing solutions to problems (Amabile, 1983; Gurteen & Associates, 1998; Yuzhu Li et al., 2011; Lin et al., 2015). It is worth noting that problem-solving required people to collaborate and share different knowledge to get work done (Aladwani, 2002; Burmeister et al., 2021), particularly when creative solutions are required in generative tasks (Gloor et al., 2007). The focus is on the business domain and technical knowledge to examine the role of informal advice networks on the problem-solving competence of software developers. Considering the pertinent role of knowledge in performing development tasks and its access to advice networks, the hypotheses developed for this study are as follows.

4.4.2 Advice Network Connectedness and Problem-Solving Competence

The basic idea of social network theory has focused on the link between social connections, brokers and knowledge resource benefits between people and organisations and what it means for network outcomes (Allen, 1984; Freeman, 1978; Wellman, 1983). McGrath et al. (2003, p. 3) assert that "*When individuals seek advice, they use interpersonal connections to fill gaps in their knowledge or experience to resolve problems or take advantage of opportunities more quickly*" (Burmeister et al., 2021). Thus, individuals seek advice to derive solutions to problems, gain meta-knowledge, reformulate problems, and validate perspectives (R. Cross et al., 2001). Knowledge itself can be described as a mix of information, experience, skills, belief, values, rules, norms, and standards that enforces the ability to act effectively. Knowledge forms can be explicit (in semantic language – coded, e.g., books, documents) and tacit (human memory – uncoded, e.g., experiences, observations, imagination, education, creativity). While the two are important, tacit knowledge cannot be easily articulated compared to explicit knowledge (Anwar et al., 2017; Aurum, Daneshgar, & Ward, 2008; Stenmark, 2002). However, social interactions are needed, more so for tacit knowledge, to share, internalise and or externalise any of this knowledge for performing tasks and deriving competitive advantage (Tassabehji et al., 2019; Wang & Wang, 2012). Addressing software development requirements requires social interactions for interpersonal connections with these knowledge types and are directly linked to business domain and technical knowledge resources among software developers (Ghobadi, 2015; Iden & Bygstad, 2018). Studies found advice sharing to be beneficial to advice-givers and advice seekers during problem-solving based on shared learning and self-evaluation to improve performance (Muthusamy & White, 2005; Shah et al., 2018; Williamson & Clark, 1989). But, to make sense of the benefits individuals' software developers accrue when they share knowledge invites exploring the impact of business domain and technical advice networks on individual problem-solving competence.

4.4.2.1 Business domain advice and problem-solving competence

Business domain advice can be described as knowledge about business processes, rules, system support, functions, and design objectives of customers that are shared (give and/or receive) to perform software tasks among software developers (Iden & Bygstad, 2018; Rus et al., 2002; Šmite et al., 2017; Tiwana, 2004). Kobayashi (2015) argues that for software developers to develop a medical system, they must first understand the practical use of the system such as listing prescriptions drug before the actual software design. Therefore, software developers must understand the business processes and customer needs before integrating technologies to deliver solutions. For example, when dealing with changes in project requirements or when new technology system support is introduced, it is argued that the ability to manage uncertainty demand relying on others for advice concerning the new workflow and software applications before jump-starting development (Nidumolu, 1995; Sykes et al., 2014). In doing so, task-related uncertainty can be reduced, and new insights can be generated for making good decisions. To drive quality work, individuals need to be competent to deal with task-related uncertainty, which arises because of the inability of individuals to generate sufficient information or knowledge required to predict a positive outcome. This is important in eliminating frustration and vulnerability caused by the low knowledge application in driving solutions to the problems (Bunger, Doogan, Hanson, & Birken, 2018; S. Nidumolu, 1995; Whelan & Conboy, 2014; Zand, 1972)

Furthermore, Lin et al. (2015) suggest that access to available domain knowledge tends to improve problem-solving competence. However, Pitts & Browne (2007) argue that not all software developers are well trained about searching for required information as well as interpreting requirements due to limited cognitive processes to drive elicitation success. But a low problem-solving effort will be required when they face familiar problems to manage the situation better (Koo & Ko, 2017). In this vein, the function of brokers as experienced knowledge agents also play a key role in ensuring that different perspectives are explored intuitively and analytically to provide answers to unresolved task problems within advice networks (Lopes & Botelho, 2013; Parise et al., 2006). Overall, the application of business domain knowledge has been considered suitable in solving problems, applying knowledge to tasks, and understanding the users (Nidumolu, 1995; Rus et al., 2002). Rus et al. (2002) suggest that a lack of business domain knowledge can lead to delay in software development effort and project success.

It is posited that individual connectedness in the advice network of individuals with diverse business domain knowledge should influence their ability to develop the required effort needed to solve business domain problems more effectively during software development. Thus, the first hypothesis (denote H1) is below:

> ***H1**: A software developer's connectedness in the business (domain) advice network will have a positive effect on their problem-solving competence.*

4.4.2.2 Technical advice and problem-solving competence

Technical advice can be described as knowledge about design, development tools, testing, debugging, programming, and understanding the coding environment that is shared (give and/or receive) to perform software tasks among software developers (Iden & Bygstad, 2018; Rus et al., 2002; Šmite et al., 2017; Tiwana, 2004). Ozer & Vogel (2015) suggest that software developers learn during problem-solving how to use certain software design tools to code and tackle software problems. Thus, individuals can freely choose who they approach for advice to solve task-related problems (Tassabehji et al., 2019). For example, studies have emphasised that software developers would need to have high problem-solving skills to define and provide solutions to technical-related quality problems (Aladwani, 2002; Balaban et al., 2015; Saltz, Yilmazel, & Yilmazel, 2016; Tekinerdogan & Aksit, 2006). In doing so, software developers would need to know who has the right knowledge when exploring diverse knowledge options required to solve specific technical problems such as fixing bugs and correcting feature errors (Conaldi & Lomi, 2013; Coughlan et al., 2003; Lin et al., 2015; Šmite et al., 2017). This is because an individual problem-solving competence to timely detect and analyse bugs will facilitate meeting commercial objectives of the organisations (Colazo, 2014). This can also inform software developers to be more knowledgeable and creative in clarifying and understanding problems where related solutions have not been earlier documented (Açıkgöz & Ö. İlhan, 2015). Although researchers have attributed competent, benevolent, integrity and systems trust or psychological climate, including mutual influence, as factors for driving collaborative behaviours to learn and share advice in joint problem-solving activities (Ebner, Leimeister, & Krcmar, 2009; Levin et al., 2002; Muthusamy, White, & White, 2005; Purvis & Zagenczyk, 2018). Similar studies found that experience of positive knowledge and reciprocal interaction, particularly face-to-face meetings with community awareness, contributes to the overall health of open-source software development practices (Naparat, Cahalane, & Finnegan, 2015). Additionally, individuals' shared traits and work history, supportive organisational culture and structure, serve as enabling factors for advice sharing required to develop problem-solving competence (Açıkgöz &

Ö. İlhan, 2015; McGrath et al., 2003; Meister, Böing, Batz, & Metternich, 2018; Siciliano, 2015).

Studies further reveal that connecting to personal networks of technical managers can serve as a great resource for knowledge assistance and opportunities to gain more confidence and competence to quickly solve difficult technical problems to maintain performance (Ehrlich & Cataldo, 2014; McGrath et al., 2003; Sykes et al., 2014). This includes managing IT deployment and addressing other software problems (Xu, Kim, & Kankanhalli, 2010). To generate non-redundant knowledge and display a creative approach to solving complex problems, individuals are expected to reach out to different others (i.e., weak ties) and not similar others (i.e., strong ties) (Granovetter, 1973; Levin et al., 2002; Perry-Smith, 2006). Although both social connections are necessary to generate new knowledge and disseminate it within the advice networks. According to structural hole theory, individuals who can bridge the gap between others as brokers tend to be more creative to benefit from and recombine diverse knowledge, in increasing problem-solving effort (Burt, 2004; Kislov et al., 2016; Whelan & Conboy, 2014). Because such individuals would be more knowledgeable to alternatively and properly deals with various software problems to improve innovation and performance (Açıkgöz & Ö. İlhan, 2015; Thomas et al., 2002). In fact, the quality of workplace contact relationships between individuals in the community of practice matters for their motivation, productivity, satisfaction and network continuity (Burmeister et al., 2021; Naparat et al., 2015; van Bel, Smolders, Ijsselsteijn, & de Kort, 2009). Finally, Gotel & Finkelstein (1994) argue that diverse techniques are needed to solve traceability problems when user requirements change during the development process. Since software developers would need to know what to do (purpose-driven), how it should be done (solution-driven), and relevant information needed (information-driven) to tackle specific problems (direction-driven). It is posited that individual connectedness in the advice network of individuals with diverse technical knowledge should influence their ability to solve technical problems more effectively during software development. Thus, the second hypothesis (denote H2) is below:

H2: *A software developer's connectedness in the technical advice network will have a positive effect on their problem-solving competence*

4.4.3 Brokerage in Advice Networks and Problem-Solving Competence

In social networks, and particularly in an organisational context, the functional role of brokers in the brokerage process of acquiring and connecting others with knowledge and informational resources, such as business domain and technical knowledge, have been linked to different network outcomes such as individual and innovation performances (Allen, 1984; Belso-Martínez et al., 2017; Obstfeld, 2005; Sawyer et al., 2010). Obstfeld et al. (2014, p. 141) define brokerage as "*behaviour by which an actor influences, manages, or facilitates interactions between other actors*." Their studies revealed that strategic brokerage action can involve sharing information or knowledge between two parties (*conduit*), where sharing of such resources may necessitate bringing such parties together to foster connections and coordination of tasks (*tertius iungens*, the third who joins). Another brokerage action involves a situation where brokers delight in keeping individuals apart to sustain their relevance by controlling knowledge resources access between the parties (*tertius gaudens or separans*, the third who enjoys) (Kwon et al., 2020; Obstfeld, 2005; Obstfeld et al., 2014). Erdogan, Karaeminogullari, Bauer, & Ellis (2020) assert that individuals who perform extra-role behaviours in providing work-related advice to the need of others are more central, influential, and approachable in the advice network.

According to Burt (2004), the understanding of brokerage revolves around gaps between two disconnected actors called structural holes. Such a brokerage position has been considered beneficial for generating new ideas for business solutions such as innovation development. However, the revisit of the brokerage process indicates that a brokerage can occur when knowledge is served between connected members rather than their disconnection – an absolute brokerage (Yuan Li et al., 2018; Obstfeld et al., 2014). Given that software knowledge requirements are complementary, and a new member of staff as a software developer might not hold them simultaneously, how he or she sources for example business knowledge to support the technical knowledge or vice versa would be important for brokering activity, particularly how it is acquired, negotiated, and integrated into software development. One of the benefits of brokers is helping to orchestrate learning processes when they connect others with tacit or implicit knowledge, which in turn, improve performance (Conklin et al., 2013; Shah et al., 2018).

Taken together, scholars have suggested that advice networks with brokers will be more effective than without them. As has been claimed that brokers can bridge communication, physical, cultural, cognitive and trust gaps in a network of relationships by facilitating knowledge interactions. This includes pointing to knowledgeable others when required in improving the problem-solving process and eventually driving solutions to work problems (Haas, 2015; Lam, 2018; Long et al., 2013; Verbeke et al., 2010). In the context of software development, studies have linked the role of brokers (e.g., gatekeeper, liaison) in facilitating business domain and technical knowledge resources to tackle software-related problems such as fixing bugs, handling sudden changes in requirements on the individual performance (Cataldo & Herbsleb, 2008; Sawyer et al., 2010), knowledge distribution (Boden & Avram, 2009; Whelan et al., 2013) and knowledge integration (Mehta & Bharadwaj, 2015; Šmite et al., 2017). Beyond software environments, other brokerage benefits include driving innovative knowledge transfer (Aalbers & Dolfsma, 2015) and knowledge expansion (Reus et al., 2020) in offline and online network studies. Therefore, a brokerage in the business domain and technical advice networks will be needed to acquire a pertinent business domain and technical knowledge required for developing software development efforts to solve various software problems (business and technical related). Hence, our third hypotheses (denote H3a/H3b) are below.

H3a: *A software developer's connectedness through a brokerage in the business advice network will have a positive effect on their problem-solving competence*

H3b: *A software developer's connectedness through a brokerage in the technical advice network will have a positive effect on their problem-solving competence*

4.4.4 The Moderating Role of Contact Quality in Advice Networks and Problem-Solving Competence

It is often said that peoples' social networks are their net worth. The quality of social contacts individuals maintain can decide their fate to succeed in organisational life because not all social contacts can create or result in meaningful positive outcomes as individuals interact together (Fasbender et al., 2020; McManus, Feyes, & Saucier, 2010). Fasbender et al. (2020, p. 409) define contact quality as "*employees' experience of positive, natural, and cooperative interactions with peers of similar status that help employees to gain socioemotional meaning at work*." These authors suggest that high-quality contacts that coworkers maintain can increase their psychological well-being and job satisfaction. For example, Loury's (2006) study of informal advice networks finds job seekers connected with highly knowledgeable high-wage workers were able to receive more quality information to help match employers, which in turn resulted in good employment offers with high compensation and better satisfaction than low-quality-referred workers (cf. Snijders et al., 2013). Similar studies have shown why personal networks with quality contacts for advice seeking have resulted in generating quality information and hence positive task impact and good performance feedback in organisations (Parker, Halgin, & Borgatti, 2015; Sasidharan et al., 2012).

Furthermore, Su & Contractor (2011) have emphasised the need for individuals (focal actors) to assess, recognise and seek knowledge expertise in others to improve their cognitive functioning in addressing their work-related problems (Burmeister et al., 2021; Parise et al., 2006). However, it is surmised that the focal actors as advice seekers must be willing to take and utilise such perspectives since it can be time and effort consuming for the more knowledgeable coworkers in providing knowledge assistance (Cross et al., 2001; Fasbender et al., 2020). While seeking knowledge in others may be warranted to attain work goals, Burmeister et al. (2021) bemoan that knowledge overload can reduce the effectiveness of the knowledge seeker. These authors suggest that lower contact quality of a co-worker will result in "*lower germane cognitive load*" and "*higher extraneous cognitive load",* respectively. But, as the quality of a contact increases, there will be a corresponding "*higher germane cognitive load*" and "*lower extraneous cognitive load*" (Burmeister et al., 2021). In other words, access to the network of quality contacts will create a positive reinforcement rather than a negative one to having a more motivated, cooperative, intelligent, and productive problem-solving interaction for higher problem-solving competence. In short, studies suggest that network affiliation (ties) to connected, competent and committed members such as brokers will be more trusted and highly instrumental in affecting individual and team performance in software development (Espinosa, Clark, & Carter, 2018; Purvis & Zagenczyk, 2018; Sasidharan et al., 2012; Sykes et al., 2014). Therefore, quality over quantity of contacts matters for deriving intrinsic and extrinsic satisfaction in interpersonal advice networks (Fasbender et al., 2020; McManus et al., 2010; van Bel et al., 2009). Hence, our fourth hypotheses (denote H4a/H4b) are below.

> **H4a**: *The positive effect of the relationship between a software developer's connectedness in the <u>business</u> advice network and problem-solving competence will be stronger when contact quality is high.*

> **H4b**: *The positive effect of the relationship between a software developer's connectedness in the <u>technical</u> advice network and problem-solving competence will be stronger when contact quality is high.*

In sum, and based on our developed hypotheses, it is posited that business advice connectedness (H1), technical advice connectedness (H2), including brokerage in business advice connectedness (H3a) and brokerage in technical advice connectedness (H4a), directedly affect an individual problem-solving competence. But that contact quality moderates the effect of both business advice connectedness (H4a) and technical advice connectedness (H4b) on problem-solving competence. Figure 14 indicates the conceptual model for this study, which shows dynamic informal advice interactions as potential network antecedents shaping the problem-solving competence of individual software developers.

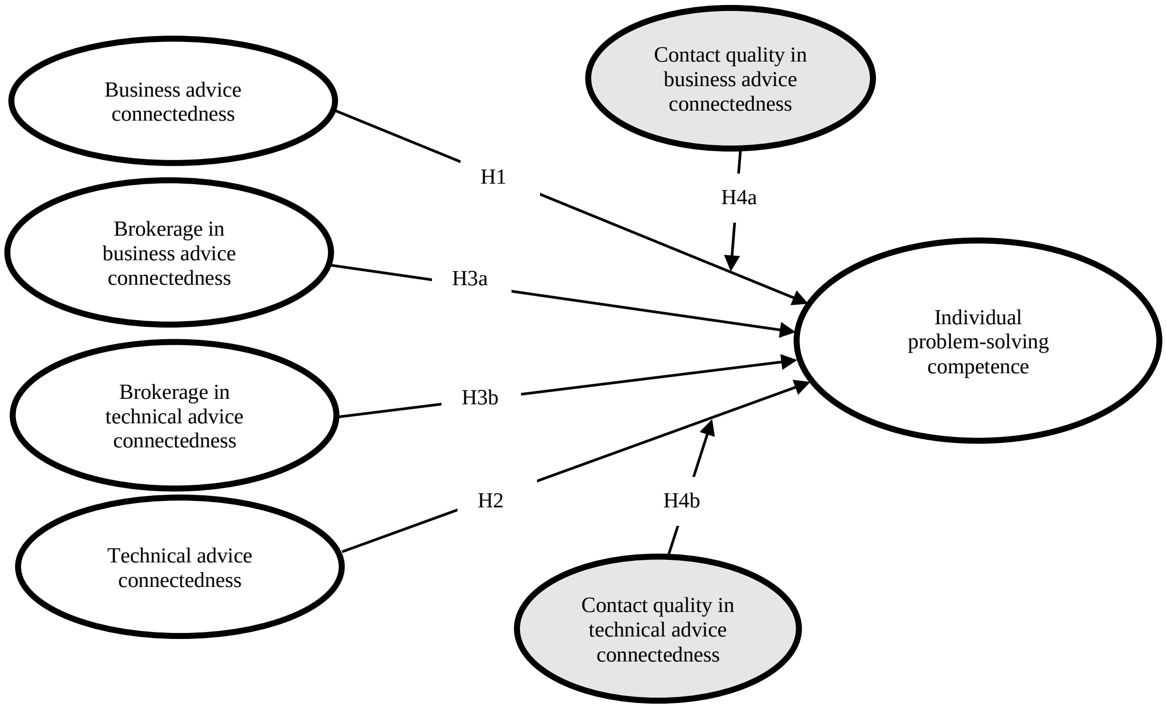


Figure 14. A conceptual model for advice network antecedents of problem-solving competence

4.5 Research Methodology

4.5.1 Data

This study applied a survey design to test the conceptual model. The use of a network survey guides in operationalising and measuring the research concepts, which is needed for building theoretical reasoning for our formulated research question. This study used data collected from software development professionals in a technology firm. A sampling frame of 370 respondents was contacted via an email online questionnaire distribution. A total of 161 responses were received, which accounted for a response rate of 43.51 per cent. A total of 8 questionnaires with incomplete responses (missing data) were excluded, leaving 153 complete questionnaires being used for our Partial Least Square Structural Equation Modelling (PLS-SEM) analysis. This data was analysed using PLS-SEM analytic technique (Smart PLS-3). The demographic characteristics in this sample indicate that 79 (49%) respondents were females while 73 (47%) respondents were males. The remaining 4% was neither belonging to male nor female (non-binary, third/prefer not to say category). The average age of the respondents was 26 years old. Most of the respondents hold bachelor's degrees and professional diplomas. The discipline or educational background of the respondents represented diverse industries namely, engineering, humanities, health sciences, natural sciences, and social sciences, with the majority operating in the field of engineering. Similarly, the respondents represented diverse ethnic backgrounds: Whites, Blacks, Asians, and Hispanics, with the most representative sample being Whites. Lastly, most of the respondents were between 1 to 4 years of professional experience. The proportion of junior software developers also far exceeded those of senior software developers.

By adhering to the PLS-SEM guidelines on the appropriateness of sample size, it is recommended that the representative sample for analysis should be 10 times the number of arrows directed at the target construct (Hair, Sarstedt, Hopkins, & Kuppelwieser, 2014; Hair, Hollingsworth, Randolph, & Chong, 2017). In this study, there are five arrows of antecedents pointing toward individual problem-solving competence. Ideally, the minimum representative sample size would be 50 respondents. However, considering the recent criticism of this approach, this study calculated an estimated minimum sample size using the G*Power 3.1 tool (Faul, Erdfelder, Buchner, & Lang, 2009; Hair, Risher, Sarstedt, & Ringle, 2019; Tehseen, Sajilan, Gadar, & Ramayah, 2017; Wirth, Maier, Laumer, & Weitzel, 2021). This was calculated based on *Linear multiple regression: Fixed model, single regression coefficient*. By setting tails of 2, an effect size (f^2) of 0.15, an α error probability of 0.05, corresponding power of 0.95 and 4 predictors. It generated 89 respondents. Thus, the current sample size of 153 is sufficient to validate statistically, the conceptual model.

4.5.2 Measures

The survey items for our constructs were developed based on a 7-point Likert scale (1= completely disagree, 7 = completely agree) to maintain consistency. Our survey items measured business advice and technical advice connectedness, a brokerage in business advice and technical advice connectedness, and problem-solving competence. *Independent variables*: Business and technical advice connectedness were measured by 3 items adapted from the social connectedness scale of measuring advice experience in social networks by van Bel, Smolders, Ijsselsteijn, & de Kort (2009). *Moderator variable*: Contact quality was measured by 3 items adapted from the contact quality scale by van Bel, Smolders, Ijsselsteijn, & de Kort (2009). Brokerage in business and technical advice connectedness were measured by 6 items adapted from tertius iungens orientation scale by Obstfeld (2005). *Dependent variable*: Problem-solving competence was measured by 8 items (4 items each measured problem-solving speed and problem-solving creativity, respectively) adapted from Atuahene-Gima & Wei (2011). Problem-solving speed to be "*the ability to find and speedily implement a large number of solutions*". While problem-solving creativity is the "*ability to discover and implement novel and cost-effective solutions*" (Atuahene-Gima & Wei, 2011, p. 83). *Control variables*: To understand possible confounding factors that can influence problem-solving competence, the control variables included age, gender, qualification, discipline, rank, ethnicity, computer self-efficacy and conscientiousness personality – this has been associated with the professionalism of individuals in getting work done. Diversity of qualifications, ethnicities and disciplines have been shown to improve a firm's level of innovativeness (Mohammadi, Broström, & Franzoni, 2017) and advice-giving interactions (Vargas & Schafer, 2013). Computer self-efficacy was measured by 4 items, and conscientiousness personality was measured by 5 items, both variables were adapted from Zhang & Venkatesh (2013). Age, gender, rank, computer self-efficacy and conscientiousness personality have been linked to affecting individual performance (Sykes et al., 2014; Zhang & Venkatesh, 2013). The details of measurements scales are provided in the Appendix.

4.6 Data Analysis and Results

4.6.1 Assessment of Measurement Model

A measurement model reveals the interacting relationship between constructs and their indicator items. In assessing a measurement model, studies suggest considering the nature of the construct, the causal direction between the latent construct and their indicator and how the indicator characteristics measure the construct (Coltman, Devinney, Midgley, & Venaik, 2008; Hair, Risher, Sarstedt, & Ringle, 2019). Given the conceptual model, it is decided to use only reflective indicators. This study first of all examined *convergent validity*, whether the model values met the acceptable threshold set by the PLS-SEM guidelines for the *indicator loadings* (≥ 0.70). Second, the *internal consistency* reliability using *composite reliability* (CR ≥ 0.70) and *Cronbach's alpha (CA)*. Third, the *convergent validity* of each construct using *average variance extracted* (AVE ≥ 0.50). Fourth, *the discriminant validity* using *Fornell-Larcker criterion* and *heterotrait-monotrait* (HTMT< 0.90) ratio (Hair et al., 2017; Hair et al., 2014; Hair et al., 2019; Henseler, Ringle, & Sarstedt, 2015).

As part of the rules of thumb in PLS-SEM guidelines, there were nine items removed (BBAC6, BTAC6, CP3, CP4, CP5, CSE1, PSC4, PSC5, PSC6) from the analysis due to low indicator (factor) loadings (<0.70). Due diligence is ensured by cross-checking to see that these items removed were captured in the remaining items of the construct, however. The reliability of the constructs was tested using Cronbach's alpha (CA) and composite reliability (CR). The CR values met the required acceptable threshold (<0.70). The Average Variance Extracted (AVE) values were also valid (>0.50). Hence, the Convergent Validity was acceptable (see Table 12). The discriminant validity was assessed using the Fornell-Larcker criterion, which shows the square root of AVE for the construct greater than the inter-construct correlations (see Table 13). The discriminant validity was also assessed using heterotrait-monotrait (HTMT) ratio of correlations, with values within the required threshold (<0.90). Hence, discriminant validity was established (see Table 13 -14).

Table 12. Loadings, Reliability, and Validity

	Loadings	CA	CR	AVE	VIF
BAC1	0.891	0.805	0.884	0.718	2.072
BAC2	0.828				
BAC3	0.821				
BBAC1	0.783	0.865	0.902	0.649	2.685
BBAC2	0.819				
BBAC3	0.821				
BBAC4	0.779				
BBAC5	0.825				
CQBAC1	0.886	0.830	0.896	0.742	2.090
CQBAC2	0.865				
CQBAC3	0.832				
TAC1	0.803	0.746	0.853	0.659	2.120
TAC2	0.817				
TAC3	0.816				
BTAC1	0.860	0.911	0.934	0.738	2.592
BTAC2	0.885				
BTAC3	0.869				
BTAC4	0.856				
BTAC5	0.825				
CQTAC1	0.878	0.853	0.906	0.764	1.983
CQTAC2	0.855				
CQTAC3	0.888				
PSC1	0.893	0.903	0.928	0.720	
PSC2	0.833				
PSC3	0.857				
PSC7	0.845				
PCS8	0.813				

Table 13. Discriminant Validity using Fornell-Larcker Criterion

	BBAC	BTAC	BAC	CQBAC	CQTAC	PSC	TAC
BBAC	*0.806*						
BTAC	0.732	*0.859*					
BAC	0.509	0.355	*0.848*				
CQBAC	-0.118	-0.074	-0.227	*0.861*			
CQTAC	-0.096	-0.039	-0.124	0.683	*0.874*		
PSC	0.754	0.633	0.521	-0.204	-0.105	*0.849*	
TAC	0.456	0.516	0.616	-0.068	-0.116	0.463	0.812

Note: Values in *italic* represent the square root of AVE

Table 14. Discriminant Validity using Heterotrait-Monotrait (HTMT) Ratio

	BBAC	BTAC	BAC	CQBAC	CQTAC	PSC	TAC
BBAC							
BTAC	0.827						
BAC	0.599	0.404					
CQBAC	0.133	0.088	0.262				
CQTAC	0.119	0.065	0.143	0.809			
PSC	0.844	0.696	0.601	0.228	0.113		
TAC	0.557	0.607	0.796	0.092	0.136	0.548	

4.6.2 Assessment of Structural Model

The structural model explains the relationship in the direction of paths between the constructs proposed in the conceptual model. It is assessed based on the required values for variance inflated factor (VIF), R^2, Q^2, f^2, root mean square residual (SRMR) and significance of paths. Our VIF values are less than 3 (see Table 12), the R^2 value is greater than 0.5, the Q^2 value is greater than 0.25, and the SRMR value is less than .10 (Hair et al., 2019; Tehseen et al., 2017). Hence, our VIF values indicate there are no multicollinearity issues. The goodness of fit of the model is based on the value of R^2 for problem-solving competence, which reflects the strength of each structural path of the model. In moderating analysis, change in the value of R^2 in explaining variance in PSC must be considered. For the base (main effect excluding the moderator) model, our explained variance of R^2 is 0.621 and 0.637 for the interaction (moderating) effect. The R^2 change of 0.016 indicates 1.6% of the additional variance. Hence, the goodness of fit increases from 62.1% to 63.7%. The effect size of f^2 values was assessed to show the weakness and strength of the latent constructs on target constructs. Given 0.016 in our additional variance of PSC, It can be stated that the introduction of two interaction terms (CQBAC/CQTAC) only has a weak effect size f^2 on PSC (see Table 15). In addition, the Q^2 value (Q^2 = .433>0) reveals and confirms the *predictive relevance* of PSC. Furthermore, our assessment of SRMR value (SRMR = .067), indicated an acceptable model fit.

Table 15. Variance in Goodness of Fit of the model

	Included	Excluded	f-squared (variance)	Effect-size
R-squared	0.637	0.621	0.016	
Effect size f^2	0.02	0.15	0.35	
	weak	moderate	strong	weak

Overall, in assessing the goodness of fit, these hypotheses were tested to ascertain the significance of the relationships between the constructs. H1 evaluates whether BAC has a significant impact on PSC. The results revealed that BAC has an insignificant impact on PSC (β =.138, t =1.870, p <.062). Hence, H1 was not supported. H2 evaluates whether TAC has a significant impact on PSC. The results revealed that TAC has an insignificant impact on PSC (β = .052, t = .675, p = .500). Hence, H2 was not supported. H3a evaluates whether BBAC has a significant impact on PSC (β =.535, t = 6.165, p <.000). The results revealed that BBAC has a significant impact on PSC. Hence, H3a was supported. H3b evaluates whether BTAC has a significant impact on PSC (β =.158, t = 2.016, p<.044).

The results indicate that BTAC has a significant impact on PSC. Hence, H3b was supported (see Table 16).

Furthermore, the direct relationship of our moderation analysis before the moderating (interaction) effect is as follows. H4a evaluates whether CQBAC has a significant relationship between BAC and PSC (β = -.146, t = 1.881, p<.060). The results revealed that CQBAC has an insignificant moderating effect between BAC and PSC. Hence. H4a was not supported. H4b evaluates whether CQTAC has a significant relationship between TAC and PSC (β =.074, t = 1.027, p <.305). The results revealed that CQTAC has an insignificant moderating effect between TAC and PSC. Hence, H4b was not supported.

Table 16. Hypotheses Testing Before Adding Moderating Effect

	β	STDEV	T Statistics	P Values (p≤0.5)	0.025	0.975
BAC -> PSC	0.138	0.074	1.870	0.062	-0.008	0.277
BBAC -> PSC	0.535	0.087	6.165	0.000	0.365	0.691
CQBAC -> PSC	-0.146	0.077	1.881	0.060	-0.304	-0.004
TAC -> PSC	0.052	0.076	0.675	0.500	-0.096	0.204
BTAC -> PSC	0.158	0.078	2.016	0.044	0.000	0.303
CQTAC -> PSC	0.074	0.073	1.027	0.305	-0.050	0.233

In this study, our bootstrapping results of 5000 resamples also generated 95% confidence intervals because confidence intervals with values different from zero indicate significant relationships. Table 18 provides the summary of our tested hypotheses. Overall, figure 15 provides the relative importance concerning the significance of weights of each indicator while their loadings determine their absolute importance.

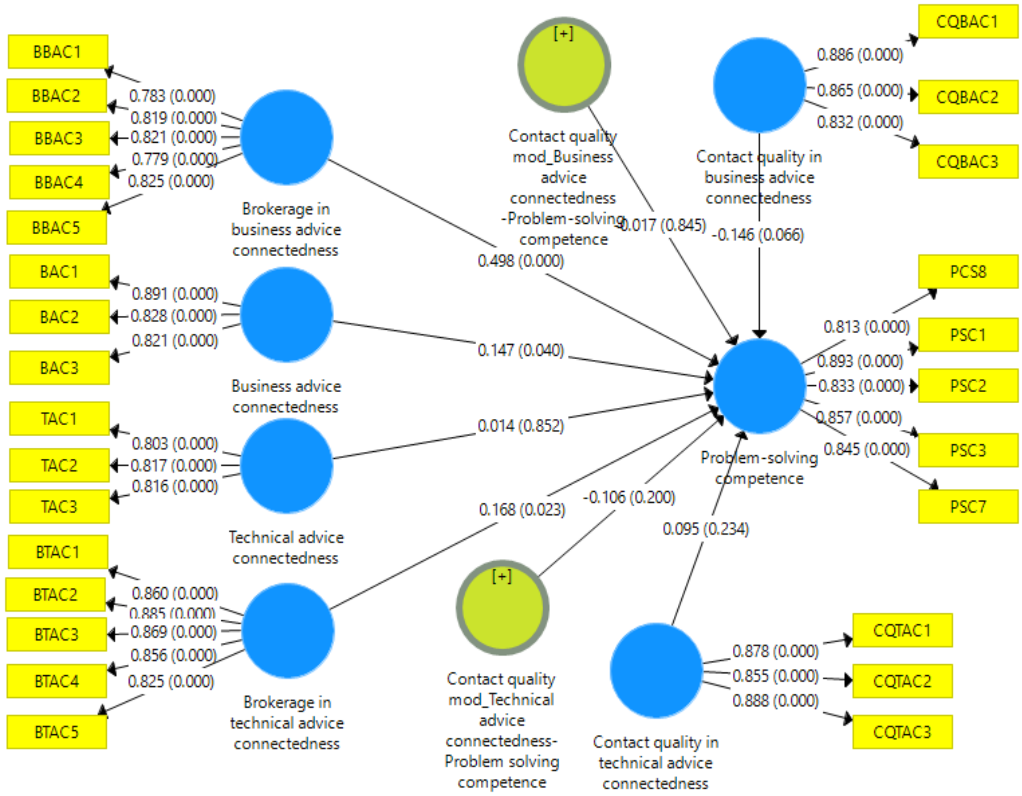


Figure 15. Assessing the level of indicator weights and significance

After controlling for the moderating effect, surprisingly, BAC has a significant impact on PSC (β = .147, t = 2.056, p = .040). While the significant impact of BBAC between BAC and PSC (β = .498, t = 5.942, p = .000), including the significant impact of BTAC between TAC and PSC (β = .168, t = 2.273, p = .023), and the insignificant impact of TAC on PSC (β = .014, t = 0.187, p = .852), remain unchanged. Thus, H1, H3a and H3b were supported, but H2 remain unsupported. Similarly, the moderating effect of CQ between BAC and PSC (β =. -.017, t = .195, p = .845), and TAC and PSC (β = -.106, t = 1.282, p = .200) to be insignificant. Thus, H4a and H4b were not supported, showing lack of any moderation. This revealed that the effect of BAC and TAC on PSC need no moderation (see Table 17 and Table 18).

Table 17. Hypotheses Testing After Adding Moderating Effect

	β	STDEV	T Statistics	P Values (p≤0.5)
BAC -> PSC	0.147	0.071	2.056	0.040
BBAC -> PSC	0.498	0.084	5.942	0.000
TAC -> PSC	0.014	0.074	0.187	0.852
BTAC -> PSC	0.168	0.074	2.273	0.023
CQ mod BAC-PSC -> PSC	-0.017	0.087	0.195	0.845
CQ mod TAC-PSC -> PSC	-0.106	0.082	1.282	0.200

Table 18. Summary of Hypotheses Results

Direct Relationship	Result	Moderation effect	Result
H1	Not supported	H1	Supported
H2	Not supported	H2	Not supported
H3a	Supported	H3a	Supported
H3b	Supported	H3b	Supported
H4a	Not supported	H4a	Not supported
H4b	Not supported	H4b	Not supported

Assessment of Importance - Performance Map Analysis (IPMA)

The importance-performance map analysis (IPMA) was calculated to identify and assess the relative importance and performance of our latent constructs and how they perform with the target construct (endogenous variables) in prioritising areas of improvement (Ringle & Sarstedt, 2016). The IPMA values show that BBAC and BTAC followed by BAC have high importance and corresponding performance scores (see Table 19). Thus, this is used to inform managerial decisions in interpreting the managerial implications for this finding as done in other research (e.g., Wirth et al., 2021).

Table 19. Importance-Performance Map Results

Construct	Importance (Total Effect)	Performance (Index Values)
BBAC	0.519	72.007
BTAC	0.149	70.458
BAC	0.14	64.491
CQBAC	-0.11	47.989
CQTAC	0.068	48.591
TAC	0.014	67.282

Table 20. Hypotheses Testing After Adding Control Variables

	β	STDEV	T Statistics	P Values (p≤0.5)
Control variables				
Age -> PSC	-0.016	0.048	0.419	0.675
Computer self-efficacy -> PSC	0.196	0.058	3.218	0.001
Conscientiousness personality -> PSC	0.226	0.070	3.369	0.001
Discipline -> PSC	-0.061	0.050	1.191	0.234
Ethnicity -> PSC	-0.036	0.062	0.560	0.576
Gender -> PSC	0.044	0.064	0.888	0.375
Qualification -> PSC	-0.031	0.051	0.527	0.598
Rank -> PSC	0.041	0.052	0.862	0.389
Tenure -> PSC	-0.018	0.055	0.240	0.811
Independent variables				
BAC -> PSC	0.073	0.071	0.836	0.403
BBAC -> PSC	0.329	0.094	3.535	0.000
TAC -> PSC	0.047	0.076	0.756	0.449
BTAC -> PSC	0.096	0.078	1.256	0.209
Moderating variables				
CQBAC -> PSC	-0.112	0.073	1.597	0.110
CQTAC -> PSC	0.039	0.071	0.752	0.452
CQ mod BAC-PSC -> PSC	0.006	0.081	0.117	0.907
CQ mod TAC-PSC -> PSC	-0.065	0.076	0.960	0.337

The PLS-SEM regression in Table 20 indicates that only two of our control variables: computer self-efficacy ($\beta = .196$, $t = 3.218$, $p = .001$) and conscientiousness personality ($\beta = .226$, $t = 3.369$, $p = .001$), were significant on PSC. Whereas only BBAC ($\beta = .329$, $t = 3.535$, $p = .000$), as part of our independent variables, maintains significance and support for hypothesis H3a. Hence, the addition of control variables has no positive effect in improving the results of our hypotheses.

4.7 Discussion and Conclusion

The purpose of this study was to empirically examine the effect of advice network structure on the problem-solving competence of software developers. This is important to establish how an individual's advice network structure shape software development effort. This study theorised that business advice and technical advice connectedness and a brokerage in business advice and technical advice connectedness for knowledge resources would enhance the problem-solving competence of software developers. But that the effect of business advice connectedness and technical advice connectedness on problem-solving competence depends on the quality of contacts within these networks. The contributions are that software developers who engage in knowledge brokering in both business and technical advice (network) connectedness will particularly increase problem-solving competence to drive their software development effort. Surprisingly, this study found that contact quality played no significant moderating role in strengthening and weakening the impact of advice networks on improving the problem-solving competence of software developers.

4.7.1 Theoretical Implications

This study provides interesting findings and implications. First, in understanding the relationship between advice networks and problem-solving competence, these findings revealed a statistically significant positive association between advice networks and individual problem-solving competence. The approach to work interactions often requires the flow of different knowledge resources, such as business advice and technical advice, as pertinent for driving the software development effort of developers. Therefore, studies focusing on either of these advice networks offer a limited understanding of their performance benefits. In this study, our multiplex network approach provides their joint advantages on the problem-solving competence of software developers. In doing so, this study adds to a stream of prior work on multiplex social interaction that outlined the combined effect of various advice network relations to be more beneficial than uniplex ones to various individual and team level tasks, including ideation and problem-solving capability (Aalbers, Dolfsma, & Koppius, 2014; Belso-Martínez et al., 2017; Shah, Parker, & Waldstrøm, 2017; Snijders et al., 2013). In addition to these prior insights, however, the research findings suggest that individuals need to engage in brokerage activities for knowledge resources to develop the required effort for amplifying their problem-solving competence. This finding makes this study one of the first to directly link business and technical advice connectedness via brokerage (as antecedents) to problem-solving competence (as outcome). This is of both scholarly and managerial relevance, as such linkage is often needed for the software development effort to tackle software-related business and technical problems. This contribution additionally extends by examining business communication networks while providing support for individual connectedness such as the role of gatekeeper in technical communication networks and the resulting positive impact on improved software development efforts and performance of distributed software development team (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014). Interestingly, the findings showed an inverse (direct) relationship when considering the impact of business advice connectedness and technical advice connectedness (without brokerage) on problem-solving competence. Except when the moderating role of contact quality resulted in a significant positive relationship between business advice connectedness (but not technical advice connectedness) and problem-solving competence while the contact quality played no significant moderating roles whatsoever.

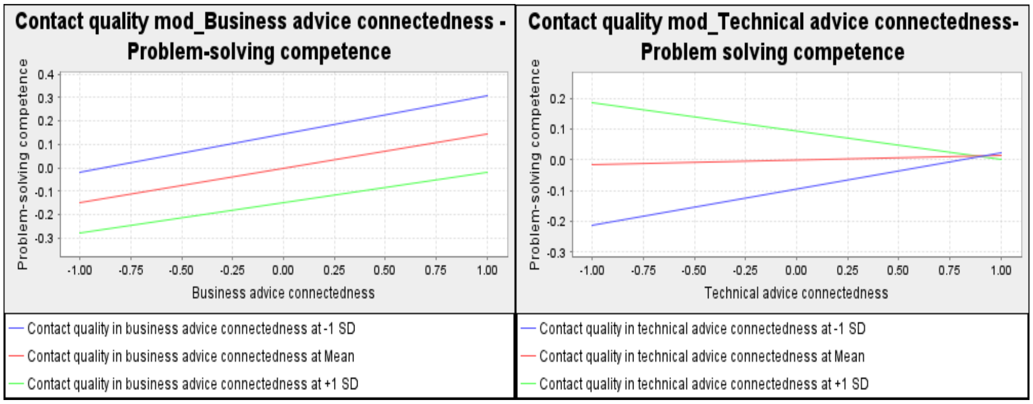


Figure 16. The moderating role of contact quality in business and technical advice connectedness on problem-solving competence

Second, though the research findings revealed there was no significant moderating role of contact quality, they nonetheless offer an interesting perspective on the role contact quality plays in converting a once negative impact of business advice connectedness (direct relationship) to drawing a positive association (moderating effect), as such advancing research contributions. In evidence of this insight, Figure 17 above shows two graph slopes for advice network sensemaking. In the business advice connectedness graph, the blue line indicates a corresponding increase in both business advice connectedness and problem-solving competence when quality contact in business advice connectedness decreases. However, counterintuitively, a decrease in quality contacts in business advice connectedness (i.e., green line) also increases business advice connectedness and problem-solving competence. In the technical advice connectedness, the blue line is pointing upward, which indicates that both technical advice connectedness and problem-solving increases when contact quality decreases. However, the green line represents the effect of high contact quality in technical advice connectedness and problem-solving competence. The green line is not flat but slightly points downwards as it goes forward, which shows that the technical advice connectedness increases and problem-solving competence decreases, albeit slightly, when contact quality in the technical advice connectedness is high. This is counterintuitive to our expectations. This finding is not consistent with a recent study by Burmeister et al. (2021), which found a significant moderating role of high contact quality in reducing knowledge overload, and in turn, increased the productivity of the knowledge seeker.

Third, these findings have significance for software development, social network, and problem-solving literature by adding to the streams of research that have considered different mechanisms other than the social network approach to understanding the factors associated with team problem-solving competence (e.g., Aladwani, 2002; Atuahene-Gima & Wei, 2011; Yuzhu Li et al., 2011). This study also provided support for previous studies that have drawn a positive relationship on problem-solving ties and network performance (Shah et al., 2018), including brokerage for business and technical knowledge sharing on other performance-related outcomes (e.g., knowledge integration) (Mehta & Bharadwaj, 2015; Sawyer et al., 2010; Šmite et al., 2017).

Fourth, and more importantly, studies suggest that social connectedness partially determines the propensity for brokerage (e.g., Aalbers, Dolfsma, & Koppius, 2013; Parise et al., 2006; Sheremata, 2000; van Bel et al., 2009). Hence, connectedness can signal the presence or absence of ties or a form of network membership; thus, an advice network can involve an exchange of resources between two individuals without brokerage playing a role. This study thereby captures this specificity of business advice connectedness and brokerage in business advice connectedness, coupled with technical advice connectedness and brokerage in technical advice connectedness. This study considered these network mechanisms for theorising as having different implications for problem-solving competence. Studies, for example, suggest that two constructs can be distinct or similar, conceptually (Hair et al., 2017). In this instance, these observed network constructs may tend to be similar but share different underlying measurements. Our convergent validity (or confirmatory factor analyses) revealed that their performance indicators loaded well (greater than 0.70) and also met the acceptable level of discriminant validity (quality robustness check). Hair et al. (2017, p. 455) suggest that "*Discriminant validity – HTMT Values lower than 0.85 for conceptually distinct constructs and below 0.90 for conceptually similar constructs; confidence intervals should not include a value of 1*".

Hence, this study has put this HTMT ratio criterion into consideration based on PLS-SEM guidelines in the literature.

Lastly, this study ran different PLS-SEM testing to establish the interrelationship or a direct link between business advice connectedness and technical advice connectedness. The results showed there is no significant difference conceptually for altering performance outcomes in these research findings. This suggests that both business and technical advice may have come into use during problem-solving activities, and their interrelationship does not matter in this conceptual model for further performance implications.

4.7.2 Managerial Implications

The findings are relevant for software practitioners. Problem-solving activity is core to software development projects as software organisations are tasked to tailor business solutions to the needs of their clients, users, and business partners. A successful project development required the right application of knowledge to solve different problems, which often depend on collaboration between software developers to work towards achieving project goals. The findings revealed that organisational leaders need to be cognisant of the advice networks for knowledge resources within their organisations and how best to promote effective networking within and between them for individual and collective knowledge resourcefulness. By promoting networking for example in both business and technical advice networks, individuals would be able to identify and utilise the diverse knowledge expertise of the more knowledgeable others, such as technical lead architects, business domain experts, or knowledge brokers. Because by being connected to the network of knowledgeable and experienced problem solvers in these networks, individuals can increase their problem-solving skills to drive better software development effort for project success. Realising the need to increase problem-solving competence demand greater brokerage and connectedness especially for less connected and isolated individuals in the advice networks for business and technical knowledge resources.

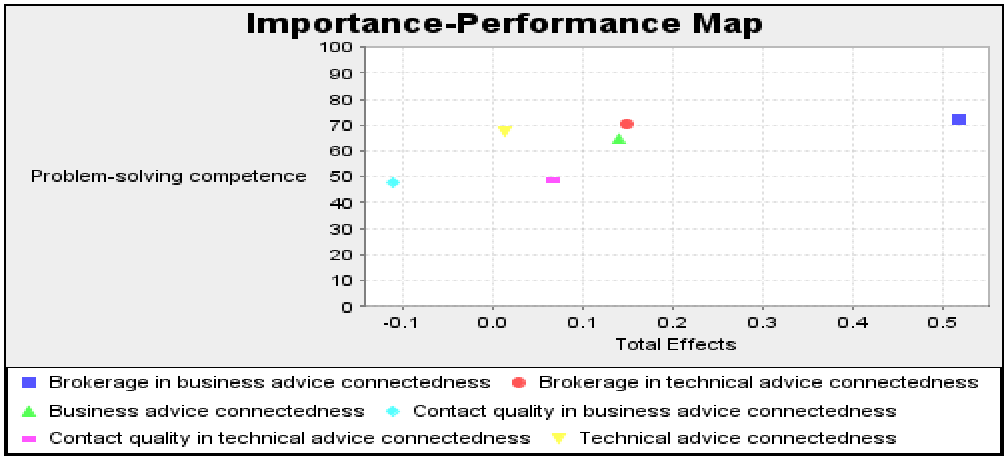


Figure 17. Importance-performance results of the network construct

Given the case study, the findings (see Figure 17) suggest that brokerage in business advice connectedness is considered of high importance and high performance for increasing problem-solving competence for successful software development. The reason is that understanding, analysing, and interpreting business problems is a necessity for matching requirements with appropriate technologies and applying technical competence to drive solutions to customer needs.

The findings further revealed that technical advice connectedness, business advice connectedness and brokerage in technical advice connectedness are of high importance yet with low performance on problem-solving competence in the software development effort. Hence, project managers need to pay more attention to these organisational knowledge mechanisms in ensuring that they are improved and optimised to deliver high performance as germane to increasing problem-solving competence require for individuals, teams, and organisational performance. In addition, this research findings revealed that contact quality is of low importance and low performance in both business and technical advice connectedness to improve problem-solving competence. While this is the least to focus on, the lack of contact quality sources for business and technical advice interactions can affect the ability to receive information quality or knowledge quality which will be needed to become creative, make timely good decisions, solve problems faster and increase productivity to meet budget and satisfy customer needs. Software development activity tends to be driven interdependently within software development teams and other stakeholders; therefore, project managers need to give attention to this in shaping problem-solving competence. It can encourage intentional expert interventions and flow of quality advice required to optimise the software development effort of their software developers.

4.8 Limitations and Future Research

The findings have a number of limitations for further research. First, our analysis was focused on the self-report measure of individual perceived experiences of their advice interactions in the whole or network level of analyses. Hence, this might have biased the results. Due to confidentiality issues and COVID-19 disruptions to work routines, there was no access to relational data. A relational data would have generated insights into the direction of ties, the level of centralisation and decentralisation, including reciprocity and brokerage roles within these advice networks, in influencing problem-solving competence. In doing so, this study would have been able to generate more contributions to the work of Sheremata (2000), where some of these network properties, including reach and free flow of information on problem-solving outcomes, were examined outside the software development domain. For example, network relationships rich in diverse yet balanced or mutual reciprocity in knowledge exchange and problem-solving interactions may facilitate greater shared understanding, and in turn, improve problem-solving competence between team members than their lack thereof. Recent studies have suggested the need for trust relationships that promote reciprocity in the exchange of knowledge to sustain healthy participation in the opensource software development effort (Naparat et al., 2015). Additional access to data on individual performance metrics set out in the organisation would provide further insights on variations in software developer networks and problem-solving performance outcomes.

Second, the lack of empirical support for part of our hypotheses such as the direct impact of business and technical advice connectedness (without brokerage) and partial support for business advice connectedness (after moderating effect of contact quality), need further investigations in similar settings. This includes understanding what contact quality means for problem-solving and performance in the software development effort of software developers. An assumption can be that contact quality will be based on the usefulness of knowledge, trust in the knowledge source, status within the organisation, network position (e.g., closing structural holes), network cohorts (being connected to well-connected others), and their availability to commit to providing knowledge assistance when called upon. Because researchers have drawn relevance to the importance of competent, benevolent and integrity-driven trust in learning and knowledge-seeking by individuals or partners (Levin et al., 2002; Muthusamy & White, 2005), as well as system trust (perceptions of the level of provisional support receivable from their organisations) (Ebner et al., 2009) or psychological climate (Purvis & Zagenczyk, 2018) where individuals operate on. This study posits that this can influence quality contact connectedness in advice networks to drive knowledge quality. This study adapted the scale for measuring contact quality proposed by van Bel et al. (2009). But due to an oversight, a recent scale by Fasbender et al. (2020) or their combinations might have predicted a different result for our moderation analysis and more empirical support. This study suggests this might alter the lack of support, though surprisingly, in these software development networks, which is beyond the scope of this study but necessary to be further investigated.

Third, studies suggest that individuals' connectedness in offline and online networks matters for knowledge resourcefulness (e.g., van Bel et al., 2009; Zhang & Venkatesh, 2013). However, the position of researchers on whether different communication networks affect performance differently is mixed in extant studies (Ding et al., 2019; Rico & Cohen, 2005; Zhang & Venkatesh, 2013). For example, Rico & Cohen (2005) suggest that individuals interacting synchronously (face-to-face/virtual meetings) and asynchronously (virtual meetings only) can experience different performance benefits. These dynamics can affect the intensity and quality of problem-solving activities and the potential to vary the problem-solving competence of software developers. Overall, this study does not capture to compare these variations about the effect of connectedness in both offline and online networks and their resulting impact on varying individual problem-solving competence.

Lastly, for the benefits of the above considerations, these findings cannot be generalised but subject to the organisational setting where this data was collected. Notwithstanding, the conceptual model need more empirical testing before it can be more validated and generalisable.

4.9 Conclusion

This study responds to calls to investigate the structure of knowledge seeking and problem-solving competence of software developers. This study revealed that business and technical advice network connectedness via brokerage for business domain and technical advice has a positive relationship with improving the problem-solving competence of software developers. But that contact quality has no moderating relationship concerning the effect of both business domain and technical advice connectedness and problem-solving competence. In sum, this study concludes that focusing on either technical or business domain advice connectedness based on uniplex network approaches does not explain the overarching network effect on problem-solving competence but by studying both networks simultaneously from a multiplex network perspective. In doing so, it gives more justification for the variance in individual problem-solving competence required for boosting software development effort, solving software problems, and producing quality software products within budgets that run more efficiently meeting requirements, and satisfying customers' commercial objectives.

Chapter 5 Contributions and Implications

5.1 Introduction and Chapter Outline

This chapter summarises the thesis research insights as guided by its purpose, objectives, findings, and implications. Section 5.2 provides the discussion and implications of the three studies in this thesis.

5.2 Discussion and Implications

This thesis investigates how the mechanisms of multiplex ties can contribute to an understanding of the structure and performance of software developers. In other words, it seeks to understand how the structure of software developers help coordinate knowledge resources and affect their work performance. In a nutshell, this thesis sought answers to the general research question: *How does the network structure of individual developers help to coordinate intra-organisational knowledge resources as a driver to performance in software development*. The sub-research questions (Q1-3) below contribute to answering the general research:

i. *How can a multiplex-based view contribute to software developers' coordination of resources towards performing software development tasks and optimising greater work outcomes? (Article 1 Related)*

ii. *How do online and offline knowledge brokering activities contribute to software development? (Article 2 Related)*

iii. *What relationship between business and technical advice networks and their brokerage activities affect a software developer problem-solving competence? (Article 3 Related)*

To answer these questions, this thesis conducted a systematic literature review to understand at a high level what multiplexity means conceptually and its different dimensions that serve as mechanisms for understanding the social network of software developers in software development settings. This was followed by pilot interviews with 6 software developers to understand their knowledge work to be investigated empirically. The first empirical work involved conducting social network analysis on 25 surveyed participants in a software organisation. The second empirical work involved a social network analysis of 153 participants within software development settings. Following these conceptual and empirical investigations, this thesis addresses a number of gaps related to the knowledge sharing processes of social network research in software development and organisational related settings.

First, and in software development, there is a multiplex problem when theoretical or empirical insights generated are limited to technical advice networks with little knowledge of business advice networks and their overlap, thereby losing data on network factors that can affect the performance of software developers (Schreiber & Zylka, 2020). Such as the effect of these structures combined rather than their separation on software development effort (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014) or problem-solving competence (Lin et al., 2015). Second, the dynamics of brokerage behaviours (e.g., for gatekeeping, coordinating, etc.) in these knowledge types in offline and online collaboration combined (Mehta & Bharadwaj, 2015; Sawyer et al., 2010; Whelan, Anderson, & Donnellan, 2015; Whelan et al., 2013) and how they culminate in better performance overall are not well understood in network context (Iden & Bygstad, 2018; Peng et al., 2013; Šmite et al., 2017; Zhang & Venkatesh, 2013). Similar unresolved issues remain in other organisational contexts (e.g., for expanding knowledge and developing innovation) in offline and online networks (Aalbers & Dolfsma, 2015; Kwon et al., 2020; Moser et al., 2017; Reus et al., 2020; Whelan et al., 2011).

Building upon the findings of these studies, this thesis provides sufficient insights to both researchers and practitioners in evaluating and managing the network (or interaction) diversity embedded in the communication profile of software developers that can alter the performance and success of software development projects. The below outlines theoretical and managerial contributions.

5.2.1 Theoretical Contributions

This thesis makes a number of *contributions* to knowledge sharing processes by exploring the social network theory of multiplexity, particularly using structural hole theory to understand the brokerage process of knowledge sharing and how they translate to the problem-solving competence of software developers. Leidner (2020) suggests that any scientific contributions will be judged from two or more factors. This includes the novelty of applying an existing theory in one field to a different context in another field, the positioning of the paper within different streams of literature, and its application to real-world problems. These factors underpin the contributions of this thesis as follows.

5.2.1.1 Multiplex-Based View of Intra-Organisational Network Relations

This thesis contributes to the understudied area of the social process of software development by providing a multiplex network understanding of the inherent potentials of diverse network ties that software developers utilise to communicate, collaborate and coordinate access to vital knowledge resources in software development settings (Schreiber & Zylka, 2020). The role of multiplex ties generate sufficient insights on the diversity and overlapping of network ties that can shape performance differentials, thus extending contributions beyond leader-follower or follower-leader ties (Peng et al., 2013), technical ties (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014) or offline-online ties (Zhang & Venkatesh, 2013) in software developer networks. Given that existing studies have investigated social network ties and impact on performance either from a communication perspective (Zhang & Venkatesh, 2013), relationship perspective (Aalbers et al., 2014; Brennecke & Rank, 2016; Cai et al., 2018) or resource perspective (Belso-Martínez et al., 2017) in the knowledge-intensive environment. While the concept of multiplexity has been readily applied in its different dimensions in other fields, this is one of the first to provide an integrated framework and its application to software development.

Rather than viewing the structures of network ties as being more homogenous whereby individuals only maintain a limited set of ties, such as maintaining formal relations in mandated collaborative work practices, such relations can also develop to overlap with informal relations between two people as multiplex ties (heterogeneous connections). These multiplex ties can also be nested with different ties, such as business and/or technical ties, offline ties and/or online ties. The various ways by which network ties can be differentiated and embedded in diverse relationships among software developers also reveal different coordinating mechanisms for communication, collaboration and maintaining multidimensional connections for resources flow. Coordination requires combining work requirements (e.g., knowledge, people, guidelines) and utilising them to drive project success. Work requirements are interpreted to be both technical and business dependencies (coordination needs). More importantly, communication is the essence of coordination. Due to the timeliness (length of time to access information and knowledge resources) and efficiency (amount and accuracy of information and knowledge accessed to the relevance of tasks) requires to fulfil coordination needs in software development (Blincoe, Valetto, & Damian, 2015; Mishra et al., 2012), software developers would need to talk to each other about their work problems to generate sufficient knowledge to solve problems and drive performance (Blincoe et al., 2015; Mishra et al., 2012; Negoita et al., 2018; Strode et al., 2012). Given that formal and informal structures, offline and online networks, and business and technical knowledge resources are considered coordination needs, hence, multiplexity enables unity of them. Therefore, an individual software developer that maintains multiplex ties will achieve better synergy of coordination needs over others with uniplex ties. This is important in achieving efficiency and effectiveness in work performance. A common slogan is that "*the whole is greater than the sum of its parts*". Thus, a multiplex network perspective generates insights into the condition by which two individuals can achieve greater tasks interdependence and knowledge generation together. Overall, MBV provides greater contributions to understanding the knowledge coordination and network dynamics of social networks in interpersonal relationships among software development teams and other knowledge workers in organisations. By considering a network governance perspective, one of the modes of software development governance is the allocation of tasks and location of knowledge of other people based on transactive knowledge systems; which is often driven by boundary-spanning activities for work coordination (Manteli, Van Den Hooff, & Van Vliet, 2014). The MBV framework thereby provides both a conceptual and empirical lens by which an individual software developer organises pertinent "structures, processes and relational mechanisms" based on informal network architectural arrangements. This informal way of network governing thus partly serves a significant purpose in coordinating and managing development requirements in such a way that addresses software development needs while also sustaining individual performance. Thus, this sheds light on how software developers and other knowledge workers, can collaborate and negotiate for resources in project development teams (Bannerman, 2009; Ben-Menahem et al., 2016; Jensen & Scacchi, 2010; Manteli et al., 2014).

5.2.1.2 Brokerage Dynamics in Interpersonal Knowledge Interactions

The multiplex-based view also forms the basis upon which software development researchers can have a holistic view of the brokerage processes of software development. In terms of the dynamic role of brokers, brokerage communication roles and knowledge interactions of software developers. The thesis findings revealed the divergence of brokerage processes as diverse brokers evolved and dissolved, in addition to the presence and absence of communication roles in the business domain and technical knowledge interactions over offline and online networks. One of the reasons for this is that brokerage ties vary among individuals with business knowledge and no technical knowledge and vice versa in offline and online networks. As there were no active brokerage functions in the online business knowledge networks coupled with the absence of the role of gatekeeper, consultant, and liaison in the online technical knowledge network. The absence of the role of itinerant or liaison was common across these networks. However, an assumption can only be made that there might not be brokerage activity in this role, which would have been a valuable addition to software development knowledge sharing potentials. Interestingly, recent studies have shown why individuals reconfigured their social contacts and ties to achieve performance. For example, a study by Wu, Antone, Srinivas, DeChurch, & Contractor (2021) found that employees that were mandated to work from home due to the COVID -19 crisis dissolved some of their intra-team ties to form and activate more inter-team ties for informational and knowledge benefits to achieve performance. Similar studies have found why a physician's networking behaviour changed during a hospital reorganisation to settle for past social ties (Giorgio et al., 2021). Other studies found that individuals' usage of technologies influenced their activation of dormant ties to manage stress and job insecurity (Yang et al., 2021). While the effect of these brokerage dynamics or network churns within this context was not correlated with individual software developer performance (lack of performance data); it nonetheless provides some insights into their brokerage, organising, and governing processes, thus advancing contributions in organisational network dynamics.

In addition, it appears that other software developers seemed to have combined multiple brokerage roles or engaged in multiplex brokering such as being a coordinator and consultant, or coordinator and representative, in facilitating technical knowledge resources within and outside their groups. Although this study cannot exactly pinpoint reasons for combining roles. But an educated guess would be associated with their knowledge and ability to perform extra functional roles, coupled with their social influence based on who knows what and why in getting work done. Kirkels & Duysters (2010, p. 377) argue, however, that "*a lack of certain roles in a network tells us something about the flow and transformation of knowledge in the field*." Based on structural holes theory, these roles would be pertinent in generating novel external technical (and business) knowledge resources and in integrating them with teams internally to drive, for example, innovation or creativity through learning for problem-solving. This in turn can promote efficiency and effectiveness in work productivity or general performance, as some studies have shown (e.g., Conklin et al., 2013; Shah et al., 2018). Hence, structural holes theory is only relevant when holes exist in network structures since the lack of brokerage roles also challenge the potential benefits of network positioning (centrality), including an individual's network behaviour.

Furthermore, and counterintuitively, studies show that it is one thing for *tertius gaudens or separans* to generate novel ideas but lack of involvement of key opinion leaders (e.g., idea connector) that would be done otherwise by *tertius iungens*, can prevent its integration for business solutions leading to commercial benefits such as developing new and marketable products (Long et al., 2013; Obstfeld, 2005; Whelan et al., 2011). Observing the direction of these *useful knowledge* interactions across communication networks provide empirical insights and extend contributions about the way individuals become relevant by appropriating value for themselves and their teams when occupying holes in network structures to initiate performance-related outcomes (e.g., innovativeness, problem-solving competence). "*Useful knowledge*" here imply knowledge that can generate quality solutions to problems. These contributions respond to numerous research calls on brokerage dynamics beyond software development by many network scholars (Aalbers & Dolfsma, 2015; Boari & Riboldazzi, 2014; Burt & Soda, 2021; Cvitanovic et al., 2017; Haas, 2015; Kwon et al., 2020; Šmite et al., 2017; Whelan et al., 2013). Hence, whether such brokerage roles are present or not reveal more about the active or inactive state of the role of individuals, knowledge-in-use and medium of interactions in the networks that are relevant in optimally or sub-optimally achieving work performance.

Overall, the result from brokerage roles means that individuals will not be able to add new knowledge or expand knowledge to solve software problems and deliver network benefits (e.g., innovativeness) when some of these key brokers (e.g., gatekeeper, liaison, consultant) are absent. This finding generates insight into the condition that may not lead to knowledge expansion and knowledge quality thereby extending recent research (Reus et al., 2020). Individuals holding such brokerage positions have also been described as power brokers and super performers as they receive and share relevant knowledge resources. Thus, the strength of each brokerage role reveals the distribution of power or power dynamics among them, especially in controlling access to vital information or knowledge quality required for performing different tasks and impacting network outcomes (e.g., Haas, 2015; Havakhor et al., 2018; Lam, 2018; Sasidharan et al., 2012). In sum, the contribution to the brokerage literature is that software developers with access to both internal and external knowledge sources for business and technical knowledge in either online and/or offline networks will contribute to greater software development performance to build on prior brokerage studies (Belso-Martínez et al., 2017; Cataldo & Herbsleb, 2008; Mehta & Bharadwaj, 2015; Sawyer et al., 2010; Whelan et al., 2013).

5.2.1.3 Brokerage in Developing Problem-Solving Competence

Following on, and considering the importance of brokerage in multiplex networks, the relationship between advice (or knowledge) networks and problem-solving competence revealed a statistically significant positive association between them. The approach to work interactions often requires the flow of different knowledge resources, such as business advice and technical advice, as pertinent for driving the software development effort of software developers. Therefore, studies focusing on either of these advice networks offer a limited understanding of their performance benefits. In Article 2, the multiplex network approach provides their joint advantages on the problem-solving competence of software developers. In doing so, this study adds to a stream of prior work on multiplex social interaction that outlined the combined effect of various advice network relations to be more beneficial than uniplex ones to various individual and team level tasks, including ideation and problem-solving capability (Aalbers, Dolfsma, & Koppius, 2014; Belso-Martínez et al., 2017; Shah, Parker, & Waldstrøm, 2017; Snijders et al., 2013). In addition to these prior insights, however, the thesis findings suggest that individuals need to engage in brokerage activities for knowledge resources to develop the required effort needed to amplify their problem-solving competence. This finding makes this study one of the first to directly link business and technical advice connectedness via brokerage (as antecedents) to problem-solving competence (as outcome). This is of both scholarly and managerial relevance, as such linkage is often needed for the software development effort to tackle business and technical problems. This contribution additionally extends by examining business communication networks while providing support for individual connectedness such as the role of gatekeeper in technical communication networks and the resulting positive impact on improved software development effort and performance of software development team (Cataldo & Herbsleb, 2008; Ehrlich & Cataldo, 2014).

Moreover, the quality of contacts matters for brokerage and social connectedness in network relationships. However, the thesis findings revealed no significant moderating role of contact quality in business and technical advice networks on the problem-solving competence of software developers. But a weak effect or variation on problem-solving competence by turning a once negative direct relationship between business advice networks and problem-solving competence into a positive interaction (but not with technical advice networks), and as such, partly advances contributions. This finding is not consistent with a recent study by Burmeister et al. (2021), which found a significant moderating role of high contact quality in reducing knowledge overload, and in turn, increased the productivity of the knowledge seeker.

Overall, and more importantly, this thesis further generates insights about the mechanisms through which individuals tend to govern their interpersonal networks based on the way they orchestrate and manage their communication structure by forming, maintaining, and changing their social contacts for a steady flow of performance-enhancing resources.

5.2.2 Managerial Contributions

The managerial contributions or contributions to practice are related to knowledge management. This is because knowledge management instruments relate to the mechanisms through which individuals and teams interact to access and apply knowledge to tasks to achieve performance.

5.2.2.1 Knowledge management

It is no surprise that knowledge remains the core asset of many organisations be it tacit or explicit because it enables them to create business value and drive competitive advantage. Interestingly, social interactions provide a springboard through which organisational actors create and transform knowledge into new product development such as software products. This makes the process of managing knowledge important for practitioner communities. Since collaborative work practices consist of a network of relationships with different knowledge interactions, the different ways through which individuals like software developers relate and are connected or linked together reveal more about their networking and collaboration tendencies and performance advantages. This thesis pays attention to the network relationships that software developers maintain to coordinate tasks and influence performance-related outcomes.

To appreciate networked organisations, leaders and managers need to consider relational diversity in work structures to know whether individuals engaging in collaborative work are maintaining both formal and informal interactions and share business and technical knowledge or advice and possibly friendship support in offline and online communications. While it is expected in a mandatory collaborative work structure that individuals assigned to teams must work together to achieve coordination needs; it is also germane to know if such interactions between two people are limited to formalised work routines or include informal interactions among software development teams. This includes understanding the diversity of knowledge resources (business, technical and personal related) being shared and communication networks connecting individuals. This is important because by knowing such variations, it will be easy to identify those who are networking, communicating, and collaborating within and across work routines or teams and whether adjustments would need to be made about the structure of tasks and communication protocols to avoid sub-optimal performance. Software development is both people and technology-driven, but the unintended consequence of silo working among Agile software developers can pose serious barriers to fostering productive work routines. Realising the different network coordination mechanisms that software developers utilise to mobilise knowledge resources can be capitalised upon to maintain balance in optimising and integrating diverse knowledge resources to heighten performance, which can separate high performing individuals and teams from moderate to low performing ones. Relationships rich in multiple resource interactions tend to have higher network benefits. Therefore, identifying individuals who are maintaining one and multiple social connections will inform management about the extent of organisational wide knowledge networking and communicational engagement with each other. To encourage effective collaboration in teams would be to employ the functional role of brokers – the like of *tertius iungens*, who are more motivated to bring people together to continue sharing resources for greater efficiency and work effectiveness.

Overall, the multiplex-based view of the relational side of the firm or organisation serves as a structural guide to managers alike about the constellation of network mechanisms in which collaborative work takes place and affect performance. It can open discussions for interventions about creating, improving, and sustaining the social infrastructure within the organisations where operational efficiency, productivity and innovation development can be sustained through formally and informally guided collaboration, communication, and coordination of work activities. More importantly, where cross-functional learning can be highly promoted and sustained to get work done quickly and effectively in driving better performance of individuals, teams, and organisations.

5.2.2.2 Achieving A Brokerage Balance, Knowledge Resourcefulness, and Problem-Solving Competence

As earlier discussed, knowledge is a key component of knowledge management practices simply because knowledge requirements are considered as production resources required to actualise business operations in knowledge-intensive organisations. Therefore, the process of knowledge sharing among software developers is of importance to practitioners in a software environment. Brokerage among software developers as knowledge workers can help organisations, especially in new product development to access relevant opportunities and flexibility needed to gain closure on work problems based on closing knowledge gaps.

Article 2 findings revealed the absence of brokers and brokerage roles in the online business knowledge sharing as well as the absence of gatekeeper, consultant, and liaison in the online technical knowledge sharing. However, there was the presence of brokers and brokerage roles in the offline business and technical knowledge sharing. Regardless of offline and online brokerage, the role of liaison was missing altogether. As stated previously, the presence and absence of roles revealed the state of knowledge flow and its transformation. Thus, missing roles can indicate a resource gap or serve as a warning signal of potential dysfunctional responsibilities of individuals to not engage one another in sharing relevant knowledge within their work roles, which might lead to inefficiency, less productivity or sub-optimal performance. For example, missing gatekeeping roles (internal) and consulting or Liaison roles (external) in knowledge sources would suggest limitations in knowledge resourcefulness and problem-solving skills needed to manage task-related uncertainty. Therefore, project managers must pay attention to the brokerage composition of software development teams whether they have a balanced communication structure in their knowledge sharing processes. Figure 18 indicates the necessity of attaining a brokerage balance –i.e., an overlap or mix of internal and external knowledge orientation in knowledge sharing processes in the offline and online networks. In doing so, this can help software developers to be more creative or sustain creativity in developing the problem-solving competence required to approach and adapt to fast changes in software development methodologies, technological innovations, and in better responding to business needs that might be needed to improve processes or produce new software products.

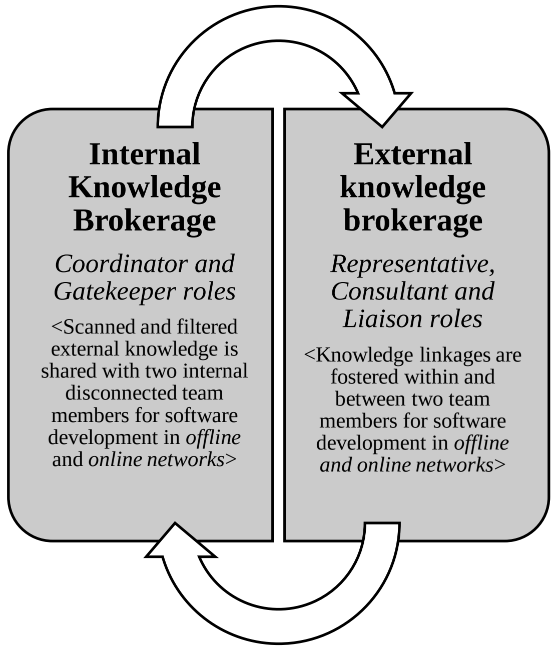


Figure 18. Achieving a brokerage balance in knowledge resourcefulness in social networks

The need to pursue a brokerage balance in knowledge resourcefulness is not only warranted in turbulent times but to drive continuous learning and growth strategies in leading innovations that are needed for long time survival in a business environment. In short, either for the benefit of software developers and senior management, holding multiple roles that combine internal and external knowledge sources in the communication networks will increase their absorptive capacity to add new knowledge resources to the existing ones. This can help create robust transactive knowledge systems for tackling wider knowledge gaps or myriad business and technical related problems during the software development process.

Furthermore, organisational managers can improve the knowledge sharing processes of the software development process when they can identify brokers holding certain roles and their personalities (agreeableness, introverts, extroverts, etc) to mobilise and support them towards achieving project objectives. An individual acting as gatekeeper and liaison between two different teams will have more social contacts across the aisle and can be instrumental in influencing organisational knowledge networking and collaboration within and between software development teams. More importantly, interpersonal interactions tend to be wired across vertical and horizontal work structures, therefore cultivating these different brokerage roles for business and technical advice is a necessity to meet the coordination needs of software developers. Because brokerage roles are not often visible but tend to emerge in self-organising networks, practitioners should be open to the service of social network analysts in helping to conduct organisational network analysis in identifying those who perform these knowledge brokering roles. Once this is done, they can begin to formally recognise them and encourage their effective functioning in facilitating knowledge connectedness with people that would otherwise not be connected in the community of practice. For example, insights from interviews revealed that software developers are encouraged to partner with others by way of knowledge sharing to innovate and implement systems design. In doing so, they are to provide peer feedback about the person who has been instrumentally helpful to them. Such peer feedback is then reviewed annually to recognise and provide incentives: awards, pay raises and bonuses.

Also, the lack of knowledge brokerage in online business knowledge sharing demands project managers to pay close attention to the underutilisation of online communication networks for work collaboration and knowledge sharing. The insights from the interviews revealed that requirements' issues are often discussed face-to-face in 15 minutes daily scrum meetings but not often virtually. This is pivotal especially when remote work becomes the new normal, and the future of responsible collaboration will largely depend on leveraging technologies to transform business processes for project development success. Therefore, organisational leaders and managers can observe their intranet discussion forums and social media platforms to understand the patterns of social interactions vis-à-vis the diverse project activities and conversations attached to posts by individuals. Identifying participating members and those leading discussions around project developments, those hanging out in online social events and those who do not participate at all can help to understand emergent brokers – in addition to officially cultivating online knowledge brokers (more knowledgeable and experienced persons across business domain and technical domains). This is important for the presiding or focal organisational managers to promote network-wide knowledge sharing among both the participating and non-participating members of the community. Therefore, online knowledge brokers (and effective emerging ones) would need to be positioned at the centre where they would be very well connected, easily reachable, and available to render functional support to peripheral specialists' others. This will certainly help trace the level of engagement of individuals whether they communicate and connect with others or not, and why such platforms can be more utilised to increase participation, collaboration, and coordination needs for achieving performance and project development success.

Additionally, while remote working may endure because of Covid-19, there is no indication that it will outlast the relevance of face-to-face offline interactions as normalcy restores. However, as the present work routines in organisations largely involve online social interactions among software development teams, paying attention to their participation, communication and collaboration activities involving online knowledge sharing practices is vital. The dynamism of brokerage activities across multiple networks also reveals why more attention should be paid to the presence and absence of some communication roles in software development knowledge sharing potentials. Recognising the different brokerage roles and those occupying them, as well as their personality and demographic attributes, can help managers and leaders on how to encourage their effective functioning and cognitive participation in knowledge sharing processes for greater collective action among software development teams. This is because some brokerage roles and individuals might be good at generating new ideas, others in connecting and developing them, while some others can potentially combine them. Insofar, an understanding of the strengths and weaknesses of each network brokerage (offline, online; coordinator, gatekeeper, etc.) can help capitalise on their combinations, usefulness, and effectiveness in software development projects.

In sum, a successful project development required the right application of knowledge to solve different problems, which often depend on collaboration between software developers to work towards achieving project goals. In this case, article 3 findings revealed that organisational leaders need to be cognisant of the advice networks for knowledge resources within their organisations and how best to promote effective networking within and between them for individual and collective knowledge resourcefulness. By promoting networking in both business and technical advice networks, individuals would be able to identify and utilise the diverse knowledge expertise of the more knowledgeable others, such as technical lead architects and business domain experts. Because by being connected to the network of knowledgeable and experienced problem solvers in these networks, individuals can increase their problem-solving competence to drive better software development performance.

Realising the need to increase problem-solving competence demand greater brokerage in advice network connectedness especially for less connected and isolated individuals in the advice networks for business and technical knowledge resources. Article 3 revealed a high level of importance and performance effect of brokerage in business advice networks in increasing the problem-solving competence of software developers. Given that information about customer needs resides in a few majorities (e.g., product owner, business domain experts), brokerage activities for business advice are often needed to understand, analyse, and interpret business problems. This is a necessity for matching requirements with appropriate technologies and applying technical competence to drive solutions to customer needs. However, the high importance and low-performance impact of business and technical advice connectedness and brokerage in technical advice connectedness on problem-solving competence deserve further attention for practitioners. This suggests that project managers need to pay more attention to these organisational knowledge mechanisms in ensuring that they are improved and optimised to deliver high performance as germane to increasing problem-solving competence. Because this has a domino effect on individuals, teams, and organisational performance. In addition, the results of low importance and low performance of the moderating role of contact quality between these advice networks and problem-solving competence further deserve attention. Because the lack of contact quality sources for business and technical advice interactions can affect the ability to receive information quality or knowledge quality which will be needed to become creative, make timely good decisions, solve problems faster and increase productivity to meet budget and satisfy customer needs. Since software development often requires individuals working interdependently within software development teams and other stakeholders; therefore, project managers need to give attention to this in shaping problem-solving competence. It can encourage intentional expert interventions and flow of quality advice required to optimise the software development effort of their software developers.

To this end, considering the relevance of these studies to innovation, practitioners can use the lens of multiplexity as a pluralistic approach in understanding different coordination needs that are often interdependent and organised around individuals and teams to create new knowledge in a self-organising manner. The need for knowledge creation is at the front burner of organisations, so the various means of organising to achieve this individually and collectively by combining multiple ties and roles through strategic networking with others into creating knowledge and solving problems together will help drive business value and achieve competitiveness in their business environments.

5.3 Limitations and Directions for Future Research

There are limitations to this thesis for further research.

Article 1: The systematic review of literature on social network studies have considered two key research string: *software development* and *information systems development*, alongside other relevant terms used in the search strategy. While such search string has generated sufficient reviews within the scope of this research, however, it does not include the term *software engineering*. Other research may consider this, as part of the search string that may be needed for more extensive reviews.

The current thesis has only considered instrumental knowledge resources: business and technical related but not friendship support empirically. The relevance of friendship ties is important in understanding resource dimensions of multiplexity (as reviewed in article 1); however, the lack of clear patterns of friendship interactions based on the interviews of software developers influenced the decision to not consider it for study in article 2 - 3. However, and retrospectively, future research may test whether this can mediate the ability to share both business and technical knowledge resources.

Article 2: Regarding the interaction between the network behaviour of a broker and structural holes opportunities in determining performance, the interesting question is: *is it the role individuals play in network structures that help in appropriating value for themselves, in terms of knowledge resourcefulness, or is it the existence of such roles that make such appropriation of value possible*? Answering this can further insight into the knowledge of structural holes as it considers the role *human agency* (i.e., personality relating to self-efficacy beliefs) plays in network behaviour and structural benefits of individuals' embeddedness in social networks (see Figure 5), as touched on in Article 1 publication). In this vein, network researchers (e.g., Grosser et al., 2019) have begun to argue that while structural holes provide a means to benefit from nonredundant information or knowledge resources, what the individual does with it matters in driving performance. This debate makes researching into the profile (e.g., *status, experience, personality, expertise)* of a broker relevant, to draw more insights undergirding the benefits of structural holes on network performance.

In like manner, in article 2, the study could not establish the different personalities that interviewees were made up of and how it influenced their propensity to interact, occupy and combine certain brokerage roles to their resource advantage. For example, in article 3 and as partly noted in article 1; the importance of personality as a link between networking behaviour or brokerage action and performance cannot be over-emphasised. According to social psychology theorists, both personality and social cognitive factors account for how people motivate and regulate their behaviour to drive their social action (or functioning) and performance in their environment. For example, personality related to "*self-monitoring*" (adaptive behaviour), "*openness to experience*" (creative potentials), "*extraversion*" (social demeanour) and "*conscientiousness*" (professionalism) have been found to shape brokering tendencies in expressive and or instrumental networks on performance (Fang et al., 2015; Meng et al., 2016; Sasovova et al., 2010; Stajkovic, Bandura, Locke, Lee, & Sergent, 2018). However, article 3 only considered conscientiousness personality as a control variable, which revealed a positive relationship between advice networks and problem-solving competence. This was tested to align with extant studies on advice networks (e.g., Sykes et al., 2014). This is an area for future research to test how some of these personality traits affect multiplex brokerage and performance.

Also, in article 2, this study has made some assumptions that might have caused some missing brokerage roles. To explore further: *do missing brokerage roles signal a lack of or unavailability of missing experts or expertise? How do these impact access to quality contacts, such that it may vary, for example, individual problem-solving competence?*

Taken together, the above points might also help in advancing understanding of what contact quality mean in software development. In article 3, and as earlier noted, this study will benefit from combining two scales in testing the role of contact quality in advice networks, particularly in software development settings. van Bel et al.'s (2009) scale reads: "*The social contacts with people in my social network feel superficial*; *My relationships with people in my social network feel superficial*; *I derive little satisfaction from my social contacts.*" But Fasbender et al.'s (2020) scale read: "*Contact with my coworkers is generally positive, negative and cooperative.*" To note, the word "*superficial*" is not explicit and so might mean different things but does not really imply whether the perceived relationship between their social contacts is negative or not cooperative. The word "*satisfaction*" can also mean a natural and cooperative relationship. Therefore, their combination can be adapted for further research. Other researchers should consider this in future studies.

5.4 Conclusion

This thesis provides a much solid network approach in which network researchers within and beyond software development settings can appreciate taking a multiplex-based view in understanding network configurations and the relational patterns of interactions undergirding performance differentials. To conclude, multiplex network configurations in this intra-organisational network context were referred to as the embedding of two individuals in a network structure with multiple ties (offline and online, formal, and informal, business, technical and friendship related) and roles (coordinating, gatekeeping, representing, liaison, and consulting related) that permit a diverse flow of knowledge and other resources – as a better driver for individual performance. In this thesis, such an interaction effect of network ties and network (brokerage) roles provide more explanatory power on various network mechanisms or antecedents shaping knowledge resourcefulness, problem-solving potentials, and ultimately problem-solving competence of software developers. The results of these studies revealed that multiplex relationships between software developers provide a greater advantage than the uniplex ones in the software development effort. Therefore, practitioners can promote and leverage interpersonal relationships rich in diverse network connectivity and brokerage capacity for knowledge and other resources in their organisations. This study offers room for future research, particularly the way multiplex-based views can inform how software developers or more general knowledge workers engage in self-organising, self-structuring and self-governing of their social networks or advice networks. In addition, the role contact quality, personality, and demographic attributes play in such a multiplex network of individuals and teams in coordinating work and achieving performance in the informal structure of organisations will be of interest for future work.

References

Aalbers, H. L., & Dolfsma, W. (2015). Bridging firm-internal boundaries for innovation: Directed communication orientation and brokering roles. *Journal of Engineering and Technology Management - JET-M*, *36*, 97–115. https://doi.org/10.1016/j.jengtecman.2015.05.005

Aalbers, R. (H L.). (2019). Rewiring the intrafirm network under downsizing: The role of tie loss on discretionary tie formation. *Long Range Planning*, (November). https://doi.org/10.1016/j.lrp.2018.11.002

Aalbers, R., Dolfsma, W., & Koppius, O. (2013). Individual connectedness in innovation networks : On the role of individual motivation. *Research Policy*, *42*(3), 624–634. https://doi.org/10.1016/j.respol.2012.10.007

Aalbers, R., Dolfsma, W., & Koppius, O. (2014). Rich ties and innovative knowledge transfer within a firm. *British Journal of Management*, *25*(4), 833–848. https://doi.org/10.1111/1467-8551.12040

Aalbers, R. H. L., Dolfsma, W., & Leenders, R. T. A. J. (2016). Vertical and Horizontal Cross-Ties: Benefits of Cross-Hierarchy and Cross-Unit Ties for Innovative Projects. *Journal of Product Innovation Management*, *33*(2). https://doi.org/10.1111/jpim.12287

Açıkgöz, A., & Ö. İlhan, Ö. (2015). Climate and Problem Solving in Software Development Teams. *Procedia - Social and Behavioral Sciences*, *207*, 502–511. https://doi.org/10.1016/j.sbspro.2015.10.120

Acuña, S. T., Gómez, M., & Juristo, N. (2009). How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology*, *51*, 627–639. https://doi.org/10.1016/j.infsof.2008.08.006

Acuña, S. T., Gómez, M. N., Hannay, J. E., Juristo, N., & Pfahl, D. (2015). Are team personality and climate related to satisfaction and software quality? Aggregating results from a twice replicated experiment. *Information and Software Technology*, *57*(1), 141–156. https://doi.org/10.1016/j.infsof.2014.09.002

Ågerfalk, P. J., Conboy, K., & Myers, M. D. (2020). Information systems in the age of pandemics: COVID-19 and beyond. *European Journal of Information Systems*, *29*(3), 203–207. https://doi.org/10.1080/0960085X.2020.1771968

Agneessens, F., & Wittek, R. (2012). Where do intra-organizational advice relations come from ? The role of informal status and social capital in social exchange. *Social Networks*, *34*(3), 333–345. https://doi.org/10.1016/j.socnet.2011.04.002

Ahuja, G., Soda, G., & Zaheer, a. (2012). Introduction to the Special Issue: The Genesis and Dynamics of Organizational Networks. *Organization Science*, *23*(2), 434–448. https://doi.org/10.1287/orsc.1110.0695

Akkerman, S. F., & Bakker, A. (2011). Boundary Crossing and Boundary Objects. *Review of Educational Research*, *81*(2), 132–169.

Aladwani, A. M. (2002). An integrated performance model of information systems projects. *Journal of Management Information Systems*, *19*(1), 185–210. https://doi.org/10.1080/07421222.2002.11045709

Albrecht, T. L., & Ropp, V. A. (1984). Communicating about innovation in networks of three US organizations. *Journal of Communication*, *34*(3), 78–91.

Aldave, A., Vara, J. M., Granada, D., & Marcos, E. (2019). Leveraging creativity in requirements elicitation within agile software development: A systematic literature review. *Journal of Systems and Software*, *157*. https://doi.org/10.1016/j.jss.2019.110396

Allen, T. J. (1984). *Managing the Flow of Technological Information Within the R&D Organization*. Cambridge, Massachusetts: MIT Press.

Amabile, T. M. (1983). The social psychology of creativity: A componential conceptualization. *Journal of Personality and Social Psychology*, *45*(2), 357–376. https://doi.org/10.1037/0022-3514.45.2.357

Antwi, S. K., & Kasim, H. (2015). Qualitative and Quantitative Research Paradigms in Business Research : A Qualitative and Quantitative Research Paradigms in Business Research : A Philosophical Reflection. *European Journal of Business and Management*, *7*(3).

Anwar, R., Rehman, M., Wang, K. S., Amin, A., & Akbar, R. (2017). Conceptual framework for implementation of knowledge sharing in global software development organizations. *IEEE Symposium on Computer Applications and Industrial Electronics*, 174–178. https://doi.org/10.1109/ISCAIE.2017.8074972

Ashworth, M. J., & Carley, K. M. (2006). Who you know vs. what you know: The impact of social position and knowledge on team performance. *Journal of Mathematical Sociology*, *30*(1), 43–75. https://doi.org/10.1080/00222500500323101

Atuahene-Gima, K., & Wei, Y. (2011). The vital role of problem-solving competence in new product success. *Journal of Product Innovation Management*, *28*(1), 81–98. https://doi.org/10.1111/j.1540-5885.2010.00782.x

Aurum, A., Daneshgar, F., & Ward, J. (2008). Investigating Knowledge Management practices in software development organisations - An Australian experience. *Information and Software Technology*, *50*(6), 511–533. https://doi.org/10.1016/j.infsof.2007.05.005

Autry, C. W., Williams, B. D., & Golicic, S. (2014). Relational and process multiplexity in vertical supply chain triads: An exploration in the U.S. restaurant industry. *Journal of Business Logistics*, *35*(1), 52–70. https://doi.org/10.1111/jbl.12034

Balaban, M., Maraee, A., Sturm, A., & Jelnov, P. (2015). A pattern-based approach for improving model quality. *Software and Systems Modeling*, *14*(4), 1527–1555. https://doi.org/10.1007/s10270-013-0390-0

Balkundi, P., Harrison, D. A., Balkundi, P., & Harrison, D. A. (2006). Ties, Leaders , and Time in Teams: Strong Inference about Network Structure's Effects on Team Viability and Performance. *Academy of Management*, *49*(1), 49–68.

Balland, P. A., Belso-Martínez, J. A., & Morrison, A. (2016). The dynamics of technical and business knowledge networks in industrial clusters: Embeddedness, status, or proximity? *Economic Geography*, *92*(1), 35–60. https://doi.org/10.1080/00130095.2015.1094370

Barthelmess, P. (2003). Collaboration and coordination in process-centered software development environments: A review of the literature. *Information and Software Technology*, *45*(13), 911–928. https://doi.org/10.1016/S0950-5849(03)00091-0

Beissel-Durrant, G. (2010). A typology of research methods within the Social Sciences. *Communication*, 1–22. Retrieved from http://eprints.soton.ac.uk/34817/

Belso-Martínez, J. A., Mas-Tur, A., & Roig-Tierno, N. (2017). Synergistic effects and the co-existence of networks in clusters. *Entrepreneurship and Regional Development*, *29*(1–2), 137–154. https://doi.org/10.1080/08985626.2016.1255429

Ben-Menahem, S. M., Von Krogh, G., Erden, Z., & Schneider, A. (2016). *Coordinating knowledge creation in multidisciplinary teams: Evidence from early-stage drug discovery*. *Academy of Management Journal* (Vol. 59). https://doi.org/10.5465/amj.2013.1214

Bernardi, M. L., Canfora, G., Di Lucca, G. A., Di Penta, M., & Distante, D. (2018). The relation between developers' communication and fix-Inducing changes: An empirical study. *Journal of Systems and Software*, *140*, 111–125. https://doi.org/10.1016/j.jss.2018.02.065

Berrett, J. L., & Holliday, B. S. (2018). The Effect of Revenue Diversification on Output Creation in Nonprofit Organizations: A Resource Dependence Perspective. *Voluntas*, *29*, 1190–1201. https://doi.org/10.1007/s11266-018-00049-5

Bhaskar, R. (1975). *A Realist Theory of Science [RTS]*. Leeds: Leeds Books (first.).

Bilecen, B., & Cardona, A. (2018). Do transnational brokers always win? A multilevel analysis of social support. *Social Networks*, *53*, 90–100. https://doi.org/10.1016/j.socnet.2017.03.001

Bird, C., Gourley, A., Devanbu, P., Gertz, M., & Swaminathan, A. (2006). Mining Email Social Networks. In *In Proceedings of the 2006 international workshop on Mining software repositories* (pp. 137–143).

Blanche, M. T., Blanche, M. J. T., Durrheim, K., & Painter, D. (2006). *Research in practice: Applied methods for the social sciences*. Juta and Company Ltd.

Bliemel, M. J., McCarthy, I. P., & Maine, E. M. A. (2014). An Integrated Approach to Studying Multiplexity in Entrepreneurial Networks. *Entrepreneurship Research Journal*, *4*(4), 367–402. https://doi.org/10.1515/erj-2014-0007

Blincoe, K., Valetto, G., & Damian, D. (2015). Facilitating coordination between software developers: A study and techniques for timely and efficient recommendations. *IEEE Transactions on Software Engineering*, *41*(10), 969–985. https://doi.org/10.1109/TSE.2015.2431680

Boari, C., & Riboldazzi, F. (2014). How knowledge brokers emerge and evolve: The role of actors' behaviour. *Research Policy*, *43*(4), 683–695. https://doi.org/10.1016/j.respol.2014.01.007

Boden, A., & Avram, G. (2009). Bridging knowledge distribution - the role of knowledge brokers in distributed software development teams. *Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering, CHASE 2009*, 8–11. https://doi.org/10.1109/CHASE.2009.5071402

Bolander, W., Satornino, C. B., Hughes, D. E., & Ferris, G. R. (2015). Social Networks Within Sales Organizations : Their Development and Importance for. *Journal of Marketing*, *79*, 1–16. https://doi.org/10.1509/jm.14.0444

Borgatti, S. P., Everett, M. G., & Freeman, L. C. (2002). Ucinet for Windows: Software for Social Network Analysis. *Harvard, MA: Analytic Technologies*.

Borgatti, S. P., & Halgin, D. S. (2011). On network theory. *Organization Science*, *22*(5), 1168–1181. https://doi.org/10.1287/orsc.1100.0641

Brennecke, J., & Rank, O. N. (2016). The interplay between formal project memberships and informal advice seeking in knowledge-intensive firms: A multilevel network approach. *Social Networks*, *44*, 307–318. https://doi.org/10.1016/j.socnet.2015.02.004

Brown, J. S., & Duguid, P. (1998). Organizing Knowledge. *California Management Review*, *40*(3), 29–46. https://doi.org/10.4324/9780203791189-4

Bunger, A. C., Doogan, N., Hanson, R. F., & Birken, S. A. (2018). Advice-seeking during implementation: A network study of clinicians participating in a learning collaborative. *Implementation Science*, *13*(1), 1–12. https://doi.org/10.1186/s13012-018-0797-7

Burmeister, A., Alterman, V., Fasbender, U., & Wang, M. (2021). Too much to know? The cognitive demands of daily knowledge seeking and the buffering role of coworker contact quality. *Journal of Applied Psychology*. https://doi.org/10.1037/apl0000925

Burt, R. (1984). Network Items and the General Social Survey. *Social Networks*, *6*, 293–339. Retrieved from http://www.sciencedirect.com/science/article/pii/0378873384900078

Burt, R. S. (2004). Structural holes and good ideas. *American Journal of Sociology*, *110*(2), 349–399. https://doi.org/10.1086/421787

Burt, R. S., Merluzzi, J., Bothner, M., Breiger, R., Frank, K., Galaskiewicz, J., … Quintane, E. (2016). Network Oscillation. *Academy of Management Discoveries*, *2*(4), 368–391.

Burt, R. S., & Soda, G. (2021). Network Capabilities: Brokerage as a Bridge Between Network Theory and the Resource-Based View of the Firm. *Journal of Management*, *47*(7), 1698–1719. https://doi.org/10.1177/0149206320988764

Buskens, V., & Van De Rijt, A. (2008). Dynamics of networks if everyone strives for structural holes. *American Journal of Sociology*, *114*(2), 371–407. https://doi.org/10.1086/590674

Çaglayan, B., & Bener, A. B. (2016). Effect of developer collaboration activity on software quality in two large scale projects. *Journal of Systems and Software*, *118*, 288–296. https://doi.org/10.1016/j.jss.2016.03.055

Cai, M., Wang, W., Cui, Y., & Stanley, H. E. (2018). Multiplex network analysis of employee performance and employee social relationships. *Physica A*, *490*, 1–12. https://doi.org/10.1016/j.physa.2017.08.008

Cataldo, M., & Herbsleb, J. D. (2008). Communication networks in geographically distributed software development. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW* (pp. 579–588). https://doi.org/10.1145/1460563.1460654

Ceci, F., & Iubatti, D. (2012). Personal relationships and innovation diffusion in SME networks : A content analysis approach. *Research Policy*, *41*(3), 565–579. https://doi.org/10.1016/j.respol.2011.10.003

Chang, K. C., Wong, J. H., Li, Y., Lin, Y. C., & Chen, H. G. (2011). External social capital and information systems development team flexibility. *Information and Software Technology*, *53*(6), 592–600. https://doi.org/10.1016/j.infsof.2011.01.007

Chen, H., Mehra, A., Tasselli, S., & Borgatti, S. P. (2021). Metwork Dynamics and Organizations: A Review and Research Agenda. *Journal of Management*, 1–115.

Cheng, N. (2009). Knowledge Sharing and Knowledge Broker of Organizations. In *2nd International Conference on Future Information Technology and Management Engineering, FITME 2009* (pp. 195–198). IEEE. https://doi.org/10.1109/FITME.2009.54

Choi, S. Y., Lee, H., & Yoo, Y. (2010). The Impact of Information Technology and Transactive Memory Systems on Knowledge Sharing, Application, and Team

Performance: A Field Study. *MIS Quarterly*, *34*(4), 855–870.

Chung, Y., & Jackson, S. E. (2013). The Internal and External Networks of Knowledge-Intensive Teams : The Role of Task Routineness. *Journal of Management*, *39*(2), 442–468. https://doi.org/10.1177/0149206310394186

Clegg, S., Josserand, E., Mehra, A., & Pitsis, T. S. (2016). The Transformative Power of Network Dynamics : A research agenda. *Organization Studies*, *37*(3), 277–291. https://doi.org/10.1177/0170840616629047

Colazo, J. (2014). Performance implications of stage-wise lead user participation in software development problem solving. *Decision Support Systems*, *67*, 100–108. https://doi.org/10.1016/j.dss.2014.08.007

Coleman, J. S. (1988). Social Capital in the Creation of Human Capital. *American Journal of Sociology*, *94*(1988), S95–S120.

Coltman, T., Devinney, T. M., Midgley, D. F., & Venaik, S. (2008). Formative versus reflective measurement models: Two applications of formative measurement. *Journal of Business Research*, *61*(12), 1250–1262. https://doi.org/10.1016/j.jbusres.2008.01.013

Conaldi, G., & Lomi, A. (2013). The dual network structure of organizational problem solving: A case study on open source software development. *Social Networks*, *35*(2), 237–250. https://doi.org/10.1016/j.socnet.2012.12.003

Conklin, J., Stolee, P., Harris, M., & Lusk, E. (2013). Knowledge brokers in a knowledge network: The case of SHRTN knowledge brokers. *Implementation Science*, *8*(7), 1–10.

Contractor, N. S., & Monge, P. R. (2003). *Theories of Communication Communication*. Oxford University Press, USA.

Contractor, N. S., Monge, P. R., & Leonardi, P. M. (2011). Multidimensional Networks and the Dynamics of Sociomateriality: Bringing Technology Inside the Network. *International Journal of Communication*, *5*, 682–720.

Coughlan, J., Lycett, M., & Macredie, R. D. (2003). Communication issues in requirements elicitation: A content analysis of stakeholder experiences. *Information and Software Technology*, *45*(8), 525–537. https://doi.org/10.1016/S0950-5849(03)00032-6

Creswell, J. W. (2007). *Qualitative Inquiry and Research Design: Choosing Among Five Approaches* (Second Edi). SAGE Publications.

Creswell, J. W. (2009). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches* (3rd Editio). SAGE Publications. https://doi.org/10.1002/tl.20234

Cross, J. N., & Cummings, J. (2004). Tie and Network Correlates of Individual Performance in Knowledge-Intensive Work. *The Academy of Management Journal*, *47*(6), 928–937.

Cross, R., Borgatti, S. P., & Parker, A. (2001). Beyond answers : dimensions of the advice network. *Social Networks*, *23*, 215–235.

Cvitanovic, C., Cunningham, R., Dowd, A. M., Howden, S. M., & van Putten, E. I. (2017). Using Social Network Analysis to Monitor and Assess the Effectiveness of Knowledge Brokers at Connecting Scientists and Decision-Makers: An Australian case study. *Environmental Policy and Governance*, *27*(3), 256–269. https://doi.org/10.1002/eet.1752

Datta, S. (2018). How does developer interaction relate to software quality? an examination of product development data. *Empirical Software Engineering*, *23*(3), 1153–1187. https://doi.org/10.1007/s10664-017-9534-0

Datta, S. (2019). Influence, Information and Team Outcomes in Large Scale Software Development. *Proceedings - Asia-Pacific Software Engineering Conference, APSEC*, 402–409. https://doi.org/10.1109/APSEC48747.2019.00061

Defranco, J. F., & Laplante, P. A. (2017). Review and Analysis of Software Development Team Communication Research. *IEEE Transactions on Professional Communication*, *60*(2), 165–182. https://doi.org/10.1109/TPC.2017.2656626

Dennis, A. R., Fuller, R. M., & Valacich, J. S. (2008). Media, Tasks, and Communication Processes: A Theory of Media Synchronicity. *MIS Quarterly*, *32*(3), 575–600.

Desmond, M., Murphy, A., Slez, A., & Lo, R. (2012). Disposable Ties and the Urban Poor. *American Journal of Sociology*, *117*(5), 1295–1335.

Ding, D., Bhattacharya, P., Phan, T. Q., & Kong, H. (2019). A Tale of Two Networks: Characterizing the Interplay between Online and Offline Communication Networks. In *Fortieth International Conference on Information Systems, Munich*

2019 (pp. 1–9).

Dissanayake, I., Dantu, R., & Nerur, S. (2013). Knowledge management in software development: The case of agile software. *19th Americas Conference on Information Systems, AMCIS 2013 - Hyperconnected World: Anything, Anywhere, Anytime*, *3*, 2298–2304.

Du, R., Ai, S., & Ren, Y. (2007). Relationship between knowledge sharing and performance: A survey in Xi'an, China. *Expert Systems with Applications*, *32*(1), 38–46. https://doi.org/10.1016/j.eswa.2005.11.001

Ebner, W., Leimeister, J. M., & Krcmar, H. (2009). Community engineering for innovations: The ideas competition as a method to nurture a virtual community for innovations. *R and D Management*, *39*(4), 342–356. https://doi.org/10.1111/j.1467-9310.2009.00564.x

Ehrlich, K., & Cataldo, M. (2014). The communication patterns of technical leaders: Impact on product development team performance. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 733–744. https://doi.org/10.1145/2531602.2531671

Eisenhardt, K. M. (1989). Building theories from case study. *The Academy of Management Review*, *14*(4), 532–550. https://doi.org/10.5465/AMR.1989.4308385

Erdogan, B., Karaeminogullari, A., Bauer, T. N., & Ellis, A. M. (2020). Perceived Overqualification at Work: Implications for Extra-Role Behaviors and Advice Network Centrality. *Journal of Management*, *46*(4), 583–606. https://doi.org/10.1177/0149206318804331

Espinosa, J. A., Clark, M. A., & Carter, D. R. (2018). Understanding Shared Familiarity and Team Performance through Network Analytics. In *Proceedings of the 51st Hawaii International Conference on System Sciences* (pp. 1–10).

Esteve Del Valle, M., & Borge Bravo, R. (2018). Leaders or Brokers? Potential Influencers in Online Parliamentary Networks. *Policy and Internet*, *10*(1), 61–86. https://doi.org/10.1002/poi3.150

Everett, M. G., & Valente, T. W. (2016). Bridging, brokerage and betweenness. *Social Networks*, *44*, 202–208. https://doi.org/10.1016/j.socnet.2015.09.001

Fang, R., Landis, B., Zhang, Z., Anderson, M. H., Shaw, J. D., & Kilduff, M. (2015).

Integrating Personality and Social Networks : A Meta-Analysis of Personality , Network Position , and Work Outcomes in Organizations. *Organization Science*, *26*(4), 1243–1260.

Fasbender, U., Burmeister, A., & Wang, M. (2020). Motivated to be socially mindful: Explaining age differences in the effect of employees' contact quality with coworkers on their coworker support. *Personnel Psychology*, *73*(3), 407–430. https://doi.org/10.1111/peps.12359

Faul, F., Erdfelder, E., Buchner, A., & Lang, A. G. (2009). Statistical power analyses using G*Power 3.1: Tests for correlation and regression analyses. *Behavior Research Methods*, *41*(4), 1149–1160. https://doi.org/10.3758/BRM.41.4.1149

Ferriani, S., Fonti, F., & Corrado, R. (2012). The social and economic bases of network multiplexity: Exploring the emergence of multiplex ties. *Strategic Organization*, *11*(1), 7–34. https://doi.org/10.1177/1476127012461576

Filiposka, S., Gajduk, A., Dimitrova, T., & Kocarev, L. (2017). Bridging online and offline social networks: Multiplex analysis. *Physica A: Statistical Mechanics and Its Applications*, *471*, 825–836. https://doi.org/10.1016/j.physa.2016.12.050

Fox, S., Leicht, R. M., & Messner, J. I. (2010). Assessing the relevance of media synchronicity theory to the use of communication media in the AECO industry. *Journal of Architectural Engineering*, *16*(2), 54–62. https://doi.org/10.1061/(ASCE)AE.1943-5568.0000012

Freeman, L. C. (1978). Centrality in social networks conceptual clarification. *Social Networks*, *1*(3), 215–239. https://doi.org/10.1016/0378-8733(78)90021-7

Fujimoto, K., & Valente, T. W. (2015). Social Science & Medicine Multiplex congruity : Friendship networks and perceived popularity as correlates of adolescent alcohol use. *Social Science & Medicine*, *125*, 173–181. https://doi.org/10.1016/j.socscimed.2014.05.023

Gallivan, M. J. (2003). The influence of software developers' creative style on their attitudes to and assimilation of a software process innovation. *Information and Management*, *40*(5), 443–465. https://doi.org/10.1016/S0378-7206(02)00039-3

Ghobadi, S. (2015). What drives knowledge sharing in software development teams: A literature review and classification framework. *Information and Management*, *52*(1), 82–97. https://doi.org/10.1016/j.im.2014.10.008

Gibbons, D. E. (2004). Friendship and Advice Networks in the Context of Changing Professional Values. *Administrative Science Quarterly*, *49*(2), 238–262.

Gimeno, J., & Woo, C. Y. (1996). Economic Multiplexity: the Structural Embeddedness of Cooperation. *Advances in Strategic Management: The Embeddedness of Strategy*, *13*, 323–361.

Giorgio, L., Mascia, D., & Cicchetti, A. (2021). Hospital reorganization and its effects on physicians' network churn: The role of past ties. *Social Science and Medicine*, *286*, 113885. https://doi.org/10.1016/j.socscimed.2021.113885

Gloor, P. A., Oster, D., Putzke, J., Fischbach, K., Schoder, D., Ara, K., … Waber, B. N. (2007). Studying microscopic peer-to-peer communication patterns. *Association for Information Systems - 13th Americas Conference on Information Systems, AMCIS 2007: Reaching New Heights*, *7*, 4391–4402. https://doi.org/10.2139/ssrn.1097111

Gómez-Solórzano, M., Tortoriello, M., & Soda, G. (2019). Instrumental and affective ties within the laboratory: The impact of informal cliques on innovative productivity. *Strategic Management Journal*, *40*(10), 1593–1609. https://doi.org/10.1002/smj.3045

Gotel, O. C. Z., & Finkelstein, A. C. W. (1994). An Analysis of the Requirements Traceability Problem. *Proceedings of the First International Conference on Requirements Engineering*, 94–101.

Gould, R. V., & Fernandez, R. M. (1989). Structures of Mediation: A Formal Approach to Brokerage in Transaction Networks. *Sociological Methodology*, *19*, 89. https://doi.org/10.2307/270949

Granovetter, M. S. (1973). Strength of Weak Ties. *American Journal of Sociology*, *78*(6), 1360–1380. https://doi.org/10.1007/sl

Greve, H. R. (2020). Learning Theory: The Pandemic Research Challenge. *Journal of Management Studies*, *57*(8), 1759–1762. https://doi.org/10.1111/joms.12631

Groenewald, T. (2004). A Phenomenological Research Design Illustrated. *International Journal of Qualitative Methods*, *3*(1), 42–55. https://doi.org/10.1177/160940690400300104

Grosser, T. J., Obstfeld, D., Labianca, G. (Joe), & Borgatti, S. P. (2019). Measuring

Mediation and Separation Brokerage Orientations: A Further Step Toward Studying the Social Network Brokerage Process. *Academy of Management Discoveries*, *5*(2), 114–136. https://doi.org/10.5465/amd.2017.0110

Gulati, R., & Gargiulo, M. (1999). Where do interorganizational networks come from? *American Journal of Sociology*, *104*(5), 1439–1493. https://doi.org/10.1086/210179

Gurteen, D., & Associates, K. (1998). Knowledge, Creativity and Innovation. *Journal of Knowledge Management*, *2*(1), 5–13.

Haas, A. (2015). Crowding at the frontier: boundary spanners, gatekeepers and knowledge brokers. *Journal of Knowledge Management*, *19*(5), 1029–1047. https://doi.org/10.1108/JKM-01-2015-0036

Hair, J., Hollingsworth, C. L., Randolph, A. B., & Chong, A. Y. L. (2017). An updated and expanded assessment of PLS-SEM in information systems research. *Industrial Management and Data Systems*, *117*(3), 442–458. https://doi.org/10.1108/IMDS-04-2016-0130

Hair, Joe F., Sarstedt, M., Hopkins, L., & Kuppelwieser, V. G. (2014). Partial least squares structural equation modeling (PLS-SEM): An emerging tool in business research. *European Business Review*, *26*(2), 106–121. https://doi.org/10.1108/EBR-10-2013-0128

Hair, Joseph F., Risher, J. J., Sarstedt, M., & Ringle, C. M. (2019). When to use and how to report the results of PLS-SEM. *European Business Review*, *31*(1), 2–24. https://doi.org/10.1108/EBR-11-2018-0203

Halevy, N., Halali, E., & Zlatev, J. J. (2019). Brokerage and brokering: An integrative review and organizing framework for third party influence. *Academy of Management Annals*, *13*(1), 215–239. https://doi.org/10.5465/annals.2017.0024

Hasan, N. (2016). Positivism : to what extent does it aid our understanding of the contemporary social world ? *Quality and Quantity*, *50*, 317–325. https://doi.org/10.1007/s11135-014-0150-4

Hassan, N. R., Mingers, J., & Stahl, B. (2018). Philosophy and information systems : where are we and where should we go ? *European Journal of Information Systems*, *27*(3), 263–277. https://doi.org/10.1080/0960085X.2018.1470776

Havakhor, T., Soror, A. A., & Sabherwal, R. (2018). Diffusion of knowledge in social media networks: effects of reputation mechanisms and distribution of knowledge roles. *Information Systems Journal*, *28*(1), 104–141. https://doi.org/10.1111/isj.12127

Haythornthwaite, C. (2001). Exploring Multiplexity : Social Network Structures in a Computer-Supported Distance Learning Class. *The Information Society*, *17*(3), 211–226. https://doi.org/10.1080/01972240152493065

Haythornthwaite, C. (2002). Strong, Weak, and Latent Ties and the Impact of New Media. *Information Society*, *18*(5), 385–401. https://doi.org/10.1080/01972240290108195

Heaney, M. T. (2014). Multiplex networks and interest group influence reputation : An exponential random graph model. *Social Networks*, *36*, 66–81. https://doi.org/10.1016/j.socnet.2012.11.003

Helms, R., & van Reijsen, J. (2008). Impact of knowledge network structure on group performance of knowledge workers in a product software company. *Proceedings of the European Conference on Knowledge Management, ECKM*, 289–296.

Henseler, J., Ringle, C. M., & Sarstedt, M. (2015). A new criterion for assessing discriminant validity in variance-based structural equation modeling. *Journal of the Academy of Marketing Science*, *43*(1), 115–135. https://doi.org/10.1007/s11747-014-0403-8

Hernandez, E., & Menon, A. (2021). Corporate Strategy and Network Change. *Academy of Management Review*, *46*(1), 80–107. https://doi.org/10.5465/AMR.2018.0013

Hite, J. M. (2005). Evolutionary Processes and Paths of Relationally Embeddded Network Ties in Emerging Entrepreneurial Firms. *Entrepreneurship Theory and Practice*, 113–144.

Hite, J. M., & Hesterly, W. S. (2001). The Evolution of Firm Networks : From Emergence to Early Growth of the Firm. *Strategic Entrepreneurship Journal*, *22*(3), 275–286.

Hoegl, M., & Parboteeah, K. P. (2007). Creativity in innovative projects: How teamwork matters. *Journal of Engineering and Technology Management - JET-M*, *24*(1–2), 148–166. https://doi.org/10.1016/j.jengtecman.2007.01.008

Hood, A. C., Cruz, K. S., & Bachrach, D. G. (2017). Conflicts with Friends: A Multiplex View of Friendship and Conflict and Its Association with Performance in Teams. *Journal of Business and Psychology*, *32*(1), 73–86. https://doi.org/10.1007/s10869-016-9436-y

Hossain, L., & Zhu, D. (2009). Social networks and coordination performance of distributed software development teams. *Journal of High Technology Management Research*, *20*(1), 52–61. https://doi.org/10.1016/j.hitech.2009.02.007

Huang, J., Baptista, J., & Newell, S. (2015). Communicational ambidexterity as a new capability to manage social media communication within organizations. *Journal of Strategic Information Systems*, *24*(2), 49–64. https://doi.org/10.1016/j.jsis.2015.03.002

Ibarra, H. (1995). Race , Opportunity , and Diversity of Social Circles in Managerial Networks. *Academy of Management Journal*, *38*(3), 673–703.

Iden, J., & Bygstad, B. (2018). The social interaction of developers and IT operations staff in software development projects. *International Journal of Project Management*, *36*(3), 485–497. https://doi.org/10.1016/j.ijproman.2017.12.001

Jiang, J. J., Klein, G., & Chang, J. Y. T. (2019). Teamwork behaviors in implementing enterprise systems with multiple projects: Results from Chinese firms. *Journal of Systems and Software*, *157*, 110392. https://doi.org/10.1016/j.jss.2019.110392

Jiang, J. J., Klein, G., Hwang, H. G., Huang, J., & Hung, S. Y. (2004). An exploration of the relationship between software development process maturity and project performance. *Information and Management*, *41*(3), 279–288. https://doi.org/10.1016/S0378-7206(03)00052-1

Jordan, G., & Segelod, E. (2006). Software innovativeness: Outcomes on project performance, knowledge enhancement, and external linkages. *R and D Management*, *36*(2), 127–142. https://doi.org/10.1111/j.1467-9310.2006.00421.x

Kadushin, C. (2002). The motivational foundation of social networks. *Social Networks*, *24*(1), 77–91. https://doi.org/10.1016/S0378-8733(01)00052-1

Kane, G. C., & Borgatti, S. P. (2011). Centrality-IS Proficiency Alignment and Workgroup Performance. *MIS Quarterly*, *35*(4), 1063–1078. https://doi.org/10.5966/sctm.2015-0279

Kane, G. C., Labianca, G. J., & Borgatti, S. P. (2014). What's Different About Social Media Networks? A Framework and Research Agenda. *MIS Quarterly*, *38*(1), 275–304.

Kauffeld-Monz, M., & Fritsch, M. (2013). Who Are the Knowledge Brokers in Regional Systems of Innovation? A Multi-Actor Network Analysis. *Regional Studies*, *47*(5), 669–685. https://doi.org/10.1080/00343401003713365

Kijkuit, B., & van den Ende, J. (2010). With a little help from our colleagues: A longitudinal study of social networks for innovation. *Organization Studies*, *31*(4), 451–479. https://doi.org/10.1177/0170840609357398

Kim, J. Y., & Goh, K. I. (2013). Coevolution and correlated multiplexity in multiplex networks. *Physical Review Letters*, *111*(5), 1–5. https://doi.org/10.1103/PhysRevLett.111.058702

Kirkels, Y., & Duysters, G. (2010). Brokerage in SME networks. *Research Policy*, *39*(3), 375–385. https://doi.org/10.1016/j.respol.2010.01.005

Kislov, R., Hodgson, D., & Boaden, R. (2016). Professionals As Knowledge Brokers: the Limits of Authority in Healthcare Collaboration. *Public Administration*, *94*(2), 472–489. https://doi.org/10.1111/padm.12227

Kitchenham, B., & Charters, S. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*. https://doi.org/10.1541/ieejias.126.589

Kleinbaum, A. M. (2016). Reorganization and Tie Decay Choices. *Management Science*, *1909*, 1–19.

Kobayashi, M. (2015). Requirement for Engineers in Embedded Software Development Outsourcing. *Annals of Business Administrative Science*, *14*(2), 97–107. https://doi.org/10.7880/abas.14.97

Koivuaho, M., & Laihonen, H. (2006). A complexity theory approach to knowledge management - Towards a better understanding of communication and knowledge flows in software development. *Electronic Journal of Knowledge Management*, *4*(1), 49–58.

Koka, B. R., Madhavan, R., Prescott, J. E., Koka, B. R., & Prescott, I. E. (2006). The Evolution of Interfirm Networks: Environment Effects on Patterns of Network Change. *The Academy of Management Review*, *31*(3), 721–737.

Koo, H. M., & Ko, I. Y. (2017). Construction and utilization of problem-solving knowledge in open source software environments. *Journal of Systems and Software*, *131*, 402–418. https://doi.org/10.1016/j.jss.2016.06.062

Kotlarsky, J., van Fenema, P. C., & Willcocks, L. P. (2008). Developing a knowledge-based perspective on coordination: The case of global software projects. *Information and Management*, *45*(2), 96–108. https://doi.org/10.1016/j.im.2008.01.001

Krackhardt, D., & Hanson, J. R. (1993). Informal Networks. *Harvard Business Review*, *71*(4), 104–111.

Kudaravalli, S., Faraj, S., & Johnson, S. L. (2017). A Configural Approach to Coordinating Expertise in Software Development Teams. *MIS Quarterly*, *41*(1), 43–64.

Kuegler, M., Smolnik, S., & Kane, G. (2015). What's in IT for employees? Understanding the relationship between use and performance in enterprise social software. *Journal of Strategic Information Systems*, *24*(2), 90–112. https://doi.org/10.1016/j.jsis.2015.04.001

Kwon, S. W., Rondi, E., Levin, D. Z., De Massis, A., & Brass, D. J. (2020). Network Brokerage: An Integrative Review and Future Research Agenda. *Journal of Management*, *46*(6), 1092–1120. https://doi.org/10.1177/0149206320914694

Labun, A., Wittek, R., & Steglich, C. (2016). The co-evolution of power and friendship networks in an organization. *Network Science*, *4*(3), 364–384. https://doi.org/10.1017/nws.2016.7

Lam, A. (2018). Boundary-crossing careers and the 'third space of hybridity': Career actors as knowledge brokers between creative arts and academia. *Environment and Planning A: Economy and Space*, *50*(8), 1716–1741. https://doi.org/10.1177/0308518X17746406

Latorre, R., & Suárez, J. (2017). Measuring social networks when forming information system project teams. *Journal of Systems and Software*, *134*, 304–323. https://doi.org/10.1016/j.jss.2017.09.019

Lazega, E., & Pattison, P. E. (1999). Multiplexity, generalized exchange and cooperation in organizations: a case study. *Social Networks*, *21*(1), 67–90.

https://doi.org/10.1016/S0378-8733(99)00002-7

Lee, A. S. (2004). Thinking about Social Theory Information Systems. *Social Theory and Philosophy for Information Systems*, (26).

Lee, J., Park, J. G., & Lee, S. (2015). Raising team social capital with knowledge and communication in information systems development projects. *International Journal of Project Management*, *33*(4), 797–807. https://doi.org/10.1016/j.ijproman.2014.12.001

Lee, S., & Lee, C. (2015). Creative Interaction and Multiplexity in Intraorganizational Networks. *Management Communication Quarterly*, *29*(1), 56–83. https://doi.org/10.1177/0893318914553120

Lee, S., & Monge, P. (2011). The Coevolution of Multiplex Communication Networks in Organizational Communities. *Journal of Communication*, *61*, 758–779. https://doi.org/10.1111/j.1460-2466.2011.01566.x

Leenders, R. T. A. J., & Dolfsma, W. A. (2016). Social Networks for Innovation and New Product Development. *Journal of Product Innovation Management*, *33*(2), 123–131. https://doi.org/10.1111/jpim.12292

Leidner, D. E. (2020). What's in a contribution? *Journal of the Association for Information Systems*, *21*(1), 238–245. https://doi.org/10.17705/1jais.00598

Levin, D. Z., Cross, R., & Abrams, L. C. (2002). The Strength of Weak Ties You Can Trust: The Mediating Role of Trust in Effective Knowledge Transfer. *Academy of Management Journal*, 1–39.

Li, Yuan, Li, N., Guo, J., Li, J., & Harris, T. B. (2018). A Network View of Advice-Giving and Individual Creativity in Teams: a Brokerage-Driven, Socially Perpetuated Phenomenon. *Academy of Management Journal*, *61*(6), 2210–2229. https://doi.org/10.5465/amj.2016.0212

Li, Yuzhu, Yang, M. H., Klein, G., & Chen, H. G. (2011). The role of team problem solving competency in information system development projects. *International Journal of Project Management*, *29*(7), 911–922. https://doi.org/10.1016/j.ijproman.2010.09.004

Licorish, S. A., & Macdonell, S. G. (2015). Communication and personality profiles of global software developers. *Information and Software Technology*, *64*, 113–131.

https://doi.org/10.1016/j.infsof.2015.02.004

Licorish, S. A., & MacDonell, S. G. (2014). Understanding the attitudes, knowledge sharing behaviors and task performance of core developers: A longitudinal study. *Information and Software Technology*, *56*(12), 1578–1596. https://doi.org/10.1016/j.infsof.2014.02.004

Licorish, S. A., & MacDonell, S. G. (2017). Exploring software developers' work practices: Task differences, participation, engagement, and speed of task resolution. *Information and Management*, *54*(3), 364–382. https://doi.org/10.1016/j.im.2016.09.005

Lin, T. C., Chen, C. M., Hsu, J. S. C., & Fu, T. W. (2015). The impact of team knowledge on problem solving competence in information systems development team. *International Journal of Project Management*, *33*(8), 1692–1703. https://doi.org/10.1016/j.ijproman.2015.07.007

Lindsjørn, Y., Sjøberg, D. I. K., Dingsøyr, T., Bergersen, G. R., & Dybå, T. (2016). Teamwork quality and project success in software development: A survey of agile development teams. *Journal of Systems and Software*, *122*, 274–286. https://doi.org/10.1016/j.jss.2016.09.028

Liu, X., Park, J., Hymer, C., & Thatcher, S. M. B. (2019). Multidimensionality: A Cross-Disciplinary Review and Integration. *Journal of Management*, *45*(1), 197–230. https://doi.org/10.1177/0149206318807285

Long, J. C., Cunningham, F. C., & Braithwaite, J. (2013). Bridges, brokers and boundary spanners in collaborative networks: a systematic review. *BMC Health Services Research*, *13*(158).

Long, Y., & Siau, K. (2008). Social network structures in open source software development teams. *Journal of Database Management*, *18*(2), 25–40. https://doi.org/10.4018/978-1-60566-172-8.ch018

Lopes, A. L., & Botelho, L. M. (2013). Distributed Coordination of Heterogeneous Agents Using a Semantic Overlay Network and a Goal-Directed Graphplan Planner. *PLoS ONE*, *8*(5). https://doi.org/10.1371/journal.pone.0062931

Lorenzen, M., & Vaarst Andersen, K. (2012). The performance effects of uniplex, multiplex and diverse social ties: A study of Bollywood film production. *The*

DRUID Society Conference 2012 on Innovation and Competitiveness - Dynamics of Organizations, Industries, Systems and Regions, 1–28.

Loury, L. D. (2006). Some contacts are more equal than others: Informal networks, job tenure, and wages. *Journal of Labor Economics*, *24*(2), 299–318. https://doi.org/10.1086/499974

MacGeorge, E. L., Feng, B., Guntzviller, L. M., & Feng, B. (2016). Advice: Expanding the Communication Paradigm. *Annals of the International Communication Association*, *40*(1), 213–243. https://doi.org/10.1080/23808985.2015.11735261

Maggioni, M. A., Breschi, S., & Panzarasa, P. (2013). Multiplexity, Growth Mechanisms and Structural Variety in Scientific Collaboration Networks. *Industry and Innovation*, *20*(3), 185–194. https://doi.org/10.1080/13662716.2013.791124

Maheshwari, M., Kumar, U., & Kumar, V. (2012). Alignment between social and technical capability in software development teams: An empirical study. *Team Performance Management: An International Journal*, *18*(1–2), 7–26. https://doi.org/10.1108/13527591211207680

Marineau, J. E., Hood, A. C., & Labianca, G. B. J. (2018). Multiplex Conflict : Examining the Effects of Overlapping Task and Relationship Conflict on Advice Seeking in Organizations. *Journal of Business and Psychology*, *33*, 595–610. https://doi.org/10.1007/s10869-017-9511-z

Marsden, P. V. (1981). Introducing Influence Processes into a System of Collective Decisions. *American Journal of Sociology*, *86*(6), 1203–1235.

Maruping, L. M., & Matook, S. (2020). The Multiplex Nature of the Customer Representative Role in Agile Information Systems Development. *MIS Quarterly*, *44*(3), 1411–1437. https://doi.org/10.25300/misq/2020/12284

Maurer, I., Bartsch, V., & Ebers, M. (2011). The value of intra-organizational social capital: How it fosters knowledge transfer, innovation performance, and growth. *Organization Studies*, *32*(2), 157–185. https://doi.org/10.1177/0170840610394301

McGrath, C. A., Vance, C. M., & Gray, E. R. (2003). With a little help from their friends: Exploring the advice networks of software entrepreneurs. *Creativity and Innovation Management*, *12*(1), 2–10. https://doi.org/10.1111/1467-8691.00262

McManus, J. L., Feyes, K. J., & Saucier, D. A. (2010). Contact and knowledge as predictors of attitudes toward individuals with intellectual disabilities. *Journal of Social and Personal Relationships*, *28*(5), 579–590. https://doi.org/10.1177/0265407510385494

Mcpherson, M., Smith-lovin, L., & Cook, J. M. (2001). Birds of A Feather: Homophily in Social Networks. *Annual Review of Sociology*, *27*, 415–444.

Mehra, A., Kilduff, M., & Brass, D. J. (2001). The Social Networks of High and Low Self-Monitors : Implications for Workplace Performance. *Administrative Science Quarterly*, *46*(1), 121–146.

Mehta, N., & Bharadwaj, A. (2015). Knowledge integration in outsourced software development: The role of sentry and guard processes. *Journal of Management Information Systems*, *32*(1), 82–115. https://doi.org/10.1080/07421222.2015.1029381

Meister, M., Böing, T., Batz, S., & Metternich, J. (2018). Problem-solving process design in production: Current progress and action required. *Procedia CIRP*, *78*, 376–381. https://doi.org/10.1016/j.procir.2018.08.316

Meng, L., Hulovatyy, Y., Striegel, A., & Milenković, T. (2016). On the interplay between individuals' evolving interaction patterns and traits in dynamic multiplex social networks. *IEEE Transactions on Network Science and Engineering*, *3*(1), 32–43. https://doi.org/10.1109/TNSE.2016.2523798

Mesch, G., & Talmud, I. (2006). The Quality of Online and Offline Relationships : The Role of Multiplexity and Duration of Social Relationships. *The Information Society*, *22*(3), 137–148. https://doi.org/10.1080/01972240600677805

Methot, J. R., Lepine, J. A., Podsakoff, N. P., & Christian, J. S. (2016). Are Workplace Friendships A Mixed Blessing? Exploring Tradeoffs Of Multiplex Relationships And Their Associations With Job Performance. *Personnel Psychology*, 311–355. https://doi.org/10.1111/peps.12109

Methot, J. R., & Rosado-solomon, E. (2019). *Multiplex relationships in organizations: Applying an ambivalence lens* (In Social). New York: Routledge.

Meyer, M. (2010). The rise of the knowledge broker. *Science Communication*, *32*(1), 118–127. https://doi.org/10.1177/1075547009359797

Min, B., Yi, S. Do, Lee, K. M., & Goh, K. I. (2014). Network robustness of multiplex networks with interlayer degree correlations. *Physical Review E*, *89*(4), 1–9. https://doi.org/10.1103/PhysRevE.89.042811

Mingers, J. (2000). The contribution of critical realism as an underpinning philosophy for OR/MS and systems. *Journal of the Operational Research Society*, *51*(11), 1256–1270. https://doi.org/10.1057/palgrave.jors.2601033

Mingers, J., & Standing, C. (2017). Why things happen – Developing the critical realist view of causal mechanisms. *Information and Organization*, *27*(3), 171–189. https://doi.org/10.1016/j.infoandorg.2017.07.001

Minor, M. J. (1983). New directions in multiplexity analysis. *Applied Network Analysis: A Methodological Introduction*, 223–244.

Mishra, D., Mishra, A., & Ostrovska, S. (2012). Impact of physical ambiance on communication, collaboration and coordination in agile software development: An empirical evaluation. *Information and Software Technology, 54*(10), 1067–1078. https://doi.org/10.1016/j.infsof.2012.04.002

Mohammadi, A., Broström, A., & Franzoni, C. (2017). Workforce Composition and Innovation: How Diversity in Employees' Ethnic and Educational Backgrounds Facilitates Firm-Level Innovativeness. *Journal of Product Innovation Management*, *34*(4), 406–426. https://doi.org/10.1111/jpim.12388

Montanari, F., Scapolan, A., & Gianecchini, M. (2016). 'Absolutely free'? The role of relational work in sustaining artistic innovation. *Organization Studies*, *37*(6), 797–821. https://doi.org/10.1177/0170840616647419

Mortensen, M., & Haas, M. R. (2018). Rethinking teams: From bounded membership to dynamic participation. *Organization Science*, *29*(2), 341–355. https://doi.org/10.1287/orsc.2017.1198

Moser, C., Groenewegen, P., & Ferguson, J. E. (2017). Meaning in organizational networks - From social to digital and back. *Research in the Sociology of Organizations*, *53*, 211–229. https://doi.org/10.1108/S0733-558X20170000053008

Muthusamy, S. K., & White, M. A. (2005). Learning and Knowledge Transfer in Strategic Alliances : A Social Exchange View. *Organization Studies*, *26*(3). https://doi.org/10.1177/0170840605050874

Muthusamy, S. K., White, M. A., & White, M. A. (2005). Learning and Knowledge

Transfer in Strategic Alliances : A Learning and Knowledge Transfer in Strategic Alliances : A Social Exchange View, (March). https://doi.org/10.1177/0170840605050874

Myers, M. D., & Newman, M. (2007). The qualitative interview in IS research: Examining the craft. *Information and Organization*, *17*(1), 2–26. https://doi.org/10.1016/j.infoandorg.2006.11.001

Nan, N., & Kumar, S. (2013). Joint effect of team structure and software architecture in open source software development. *IEEE Transactions on Engineering Management*, *60*(3), 592–603. https://doi.org/10.1109/TEM.2012.2232930

Naparat, D., Cahalane, M., & Finnegan, P. (2015). Healthy community and healthy commons: “Opensourcing” as a sustainable model of software production. *Australasian Journal of Information Systems*, *19*, 1–13. https://doi.org/10.3127/ajis.v19i0.1221

Negoita, B., Lapointe, L., & Rivard, S. (2018). Collective information systems use: A typological theory. *MIS Quarterly: Management Information Systems*, *42*(4), 1281–1301. https://doi.org/10.25300/MISQ/2018/13219

Newig, J., Günther, D., & Pahl-Wostl, C. (2010). Synapses in the Network : Learning in Governance Networks in the. *Ecology and Society*, *15*(4), 24.

Ngwenyama, O., & Lee, S. (1997). Communication Richness in Electronic Mail : Critical Social Theory and the Contextuality of Meaning. *MIS Quarterly*, *21*(2), 145–168.

Nichols, W. R. (2006). Using TSP and Effective Communication Networks More Active Communication with Role Manager A : Star B : Web. *Cross Talk, The Journal of Defense Software Engineering*, (January), 11–15.

Nidumolu, S. (1995). The effect of coordination and uncertainty on software project performance: Residual performance risk as an intervening variable. *Information Systems Research*, *6*(3), 191–219. https://doi.org/10.1287/isre.6.3.191

Nidumolu, S. R., & Knotts, G. W. (1998). The effects of customizability and reusability on perceived process and competitive performance of software firms. *MIS Quarterly*, *22*(2), 105–128. https://doi.org/10.2307/249392

Nootjarat, R., Chantatub, W., & Chongstitvatana, P. (2015). The Moderating Effect of

Leader Centrality on Team Cohesion and Performance in Software Development Projects. *International Journal of Business and Information*, *10*(3), 295.

O Connor, M., Conboy, K., & Dennehy, D. (2021). COVID-19 affected remote workers: a temporal analysis of information system development during the pandemic. *Journal of Decision Systems*, 1–27. https://doi.org/10.1080/12460125.2020.1861772

Obstfeld, D. (2005). Social Networks , the Tertius Iungens Orientation , and Involvement in Innovation. *Administrative Science Quarterly*, *50*(1), 100–130.

Obstfeld, D., Borgatti, S. P., & Davis, J. (2014a). Brokerage as a process: Decoupling third party action from social network structure. *Research in the Sociology of Organizations*, *40*(July), 135–159. https://doi.org/10.1108/S0733-558X(2014)0000040007

Obstfeld, D., Borgatti, S. P., & Davis, J. (2014b). Brokerage as a process: Decoupling third party action from social network structure. *Research in the Sociology of Organizations*, *40*, 135–159. https://doi.org/10.1108/S0733-558X(2014)0000040007

Oh, I., Charlier, S. D., Mount, M. K., & Berry, C. M. (2014). The two faces of high self-monitors : Chameleonic moderating effects of self-monitoring on the relationships between personality traits and counterproductive work behaviors. *Journal of Organizational Behaviour*, *35*, 92–111. https://doi.org/10.1002/job

Okoli, C., & Schabram, K. (2010). *Working Papers on Information Systems A Guide to Conducting a Systematic Literature Review of Information Systems Research*. *Working Papers on Information Systems* (Vol. 10). https://doi.org/10.2139/ssrn.1954824

Ong, A., Tan, G. W., & Kankanhalli, A. (2005). Team Expertise and Performance in Information Systems Development Projects. *Proceedings of the 9th Asia Pacific Conference on Information Systems, Bangkok, Thailand*, (June), 1456–1462.

Ozer, M., & Vogel, D. (2015). Contextualized Relationship between Knowledge Sharing and Performance in Software Development. *Journal of Management Information Systems*, *32*(2), 134–161. https://doi.org/10.1080/07421222.2015.1063287

Packard, M. D. (2017). Where did interpretivism go in the theory of entrepreneurship? *Journal of Business Venturing*, *32*(5), 536–549. https://doi.org/10.1016/j.jbusvent.2017.05.004

Parise, S., Cross, R., & Davenport, T. H. (2006). Strategies for preventing a knowledge-loss crisis. *MIT Sloan Management Review*, *47*(4), 31–38.

Park, J. G., & Lee, J. (2014). Knowledge sharing in information systems development projects: Explicating the role of dependence and trust. *International Journal of Project Management*, *32*(1), 153–165. https://doi.org/10.1016/j.ijproman.2013.02.004

Park, J., & Lee, J. (2017). Trust and Dependence Model of Knowledge Sharing in Systems Development. In *Proceedings of the Thirty Eighth International Conference on Information Systems (ICIS)* (pp. 1–19). Retrieved from http://aisel.aisnet.org/icis2017http://aisel.aisnet.org/icis2017/IS-Development/Presentations/13

Parker, A., Halgin, D. S., & Borgatti, S. P. (2015). Dynamics of Social Capital: Effects of Performance Feedback on Network Change. *Organization Studies*, *37*(3), 375–397. https://doi.org/10.1177/0170840615613371

Parker, A., Halgin, D. S., & Borgatti, S. P. (2016). Dynamics of Social Capital: Effects of Performance Feedback on Network Change. *Organization Studies*, *37*(3), 375–397. https://doi.org/10.1177/0170840615613371

Pavan, E. (2014). Embedding digital communications within collective action networks: a multidimensional network approach. *Mobilization: An International Quarterly*, *19*(4), 441–455.

Payne, K. C., Keith, M. J., Babb, J., & Spruill, A. N. (2018). Development and Validation of the Information Systems Creative-Self-Efficacy Scale. *Proceedings of the 51st Hawaii International Conference on System Sciences*. https://doi.org/10.24251/hicss.2018.017

Peng, G., Wan, Y., & Woodlock, P. (2013). Network ties and the success of open source software development. *Journal of Strategic Information Systems*, *22*(4), 269–281. https://doi.org/10.1016/j.jsis.2013.05.001

Perry-Smith, J. E. (2006). Social yet creative: The role of social relationships in facilitating individual creativity. *Academy of Management Journal*, *49*(1), 85–101. https://doi.org/10.5465/AMJ.2006.20785503

Peters, L. D., Johnston, W. J., Pressey, A. D., & Kendrick, T. (2010). Collaboration and collective learning : networks as learning organisations. *Journal of Business & Industrial Marketing*, *25*(6), 478–484. https://doi.org/10.1108/08858621011066062

Pitts, M. G., & Browne, G. J. (2007). Improving requirements elicitation: an empirical investigation of procedural prompts. *Information Systems Journal*, *17*, 89–110.

Podolny, J. M., & Baron, J. N. (1997). Resources and relationships: Social networks and mobility in the workplace. *American Sociological Review*, *62*(5), 673–693.

Pollock, T. G., Porac, J. F., & Wade, J. B. (2004). Constructing deal networks: Brokers as network "architects" in the U.S. IPO market and other examples. *Academy of Management Review*, *29*(1), 50–72. https://doi.org/10.5465/AMR.2004.11851710

Purvis, R., & Zagenczyk, T. (2018). The Development of Psychological Climate Within Information Systems Projects and Its Impact on Project Performance. In *European Conference on Information Systems* (pp. 1–15).

Rasch, R. H., & Tosi, H. L. (1992). Factors affecting software developers' performance: An integrated approach. *MIS Quarterly: Management Information Systems*, *16*(3), 395–409. https://doi.org/10.2307/249535

Reus, B., Moser, C., & Groenewegen, P. (2020). Expanding organisational knowledge online: the role of bridging members in knowledge expansion in online groups. *Innovation: Organization and Management*, 1–24. https://doi.org/10.1080/14479338.2020.1840381

Rico, R., & Cohen, S. G. (2005). Effects of task interdependence and type of communication on performance. *Journal of Managerial Psychology*, *20*(3–4), 261–274. https://doi.org/10.1108/02683940510589046

Ringle, C. M., & Sarstedt, M. (2016). Gain more insight from your PLS-SEM results the importance-performance map analysis. *Industrial Management and Data Systems*, *116*(9), 1865–1886. https://doi.org/10.1108/IMDS-10-2015-0449

Roberts, K. H., & O'Reilly, C. A. (1979). Some Correlations of Communication Roles in Organizations. *Academy of Management Journal*, *22*(1), 42–57. https://doi.org/10.5465/255477

Robillard, P. N., & Robillard, M. P. (2000). Types of collaborative work in software engineering. *Journal of Systems and Software*, *53*(3), 219–224. https://doi.org/10.1016/S0164-1212(00)00013-3

Rogan, M. (2014). Executive Departures Without Client Losses: the Role of Multiplex Ties in Exchange Partner Retention. *Academy of Management Journal*, *57*(2), 563. Retrieved from http://proxy.kennesaw.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=edb&AN=95610053&site=eds-live&scope=site

Rogers, E. M., & Kincaid, L. (1981). Communication networks: Towards a new paradigm for research. *Free Press*.

Rondeau, P. J., Ragu-Nathan, T. S., & Vonderembse, M. A. (2006). How involvement, IS management effectiveness, and end-user computing impact IS performance in manufacturing firms. *Information and Management*, *43*, 93–107. https://doi.org/10.1016/j.im.2005.02.001

Rosenkopf, L., & Padula, G. (2008). Investigating the Microstructure of Network Evolution: Alliance Formation in the Mobile Communications Industry. *Organization Science*, *19*(5), 669–687. https://doi.org/10.1287/orsc.1070.0339

Rothman, N. B., Pratt, M. G., Rees, L., & Vogus, T. J. (2017). Understanding the Dual Nature of Ambivalence: Why and When Ambivalence Leads to Good and Bad Outcomes. *Academy of Management Annals*, *11*(1), 33–72. https://doi.org/10.5465/annals.2014.0066

Rowley, T. J., Greve, H. R., Rao, H., Baum, J. A. C., & Shipilov, A. V. (2005). Time to Break Up : Social and Instrumental Antecedents of Firms Exits from Exchange Cliques. *Academy of Management*, *48*(3), 499–520.

Rus, I., Lindvall, M., & Sinha, S. S. (2002). Knowledge management in software engineering. *IEEE Software*, *19*(3), 26–38. https://doi.org/10.1109/MS.2002.1003450

Ryan, S., & O’Connor, R. V. (2013). Acquiring and Sharing tacit knowledge in software development teams: An empirical study. *Information and Software Technology*, *55*(9), 1614–1624. https://doi.org/10.1016/j.infsof.2013.02.013

Saltz, J. S., Yilmazel, S., & Yilmazel, O. (2016). Not all software engineers can become good data engineers. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, 2896–2901. https://doi.org/10.1109/BigData.2016.7840939

Sarker, S., Ahuja, M., Sarker, S., & Kirkeby, S. (2011). The Role of Communication and Trust in Global Virtual Teams: A Social Network Perspective. *Journal of Management Information Systems*, *28*(1), 273–310. https://doi.org/10.2753/MIS0742-1222280109

Sasidharan, S., Santhanam, R., Brass, D. J., & Sambamurthy, V. (2012). The Effects of Social Network Structure on Enterprise Systems Success : A Longitudinal Multilevel Analysis. *Information Systems Research*, *23*(3), 658–678.

Sasovova, Z., Mehra, A., Borgatti, S. P., & Schippers, M. C. (2010). Network Churn: The Effects of Self-Monitoring Personality on Brokerage Dynamics. *Administrative Science Quarterly*, *55*(4), 639–670. https://doi.org/10.2189/asqu.2010.55.4.639

Saunders, M., Lewis, P., & Thornhill, A. (2009). *Research Methods for Business Students*. *Research methods for business students*.

Sawyer, S., Guinan, P. J., & Cooprider, J. (2010). Social interactions of information systems development teams : a performance perspective. *Information Systems Journal*, *20*, 81–107. https://doi.org/10.1111/j.1365-2575.2008.00311.x

Schlierf, K., & Meyer, M. (2013). Situating knowledge intermediation: Insights from science shops and knowledge brokers. *Science and Public Policy*, *40*(4), 430–441. https://doi.org/10.1093/scipol/sct034

Schmidt, C. T., Kude, T., Heinzl, A., & Mithas, S. (2014). How agile practices influence the performance of software development teams: The role of shared mental models and backup. *35th International Conference on Information Systems "Building a Better World Through Information Systems", ICIS 2014*, 1–18.

Schreiber, R. R., & Zylka, M. P. (2020). *Social Network Analysis in Software Development Projects: A Systematic Literature Review*. *International Journal of Software Engineering and Knowledge Engineering* (Vol. 30). https://doi.org/10.1142/S021819402050014X

Shah, N. P., Cross, R., & Levin, D. Z. (2018). *Performance Benefits From Providing Assistance in Networks: Relationships That Generate Learning*. *Journal of*

Management (Vol. 44). https://doi.org/10.1177/0149206315584822

Shah, N. P., Parker, A., & Waldstrøm, C. (2017). Examining the Overlap: Individual Performance Benefits of Multiplex Relationships. *Management Communication Quarterly*, *31*(1), 5–38. https://doi.org/10.1177/0893318916647528

Sheremata, W. A. (2000). Centrifugal and Centripetal Forces in Radical New Product Development under Time. *Academy of Management*, *25*(2), 389–408.

Shipilov, A. (2012). Strategic multiplexity. *Strategic Organization*, *10*(3), 215–222. https://doi.org/10.1177/1476127012452825

Shipilov, A., Gulati, R., Kilduff, M., Li, S., & Tsai, W. (2014). Relational Pluralism Within And Between Organizations. *Academy of Management Journal*, *57*(2), 449–459.

Shipilov, A., Rowley, T. J., & Aharonson, B. (2006). When do Networks Matter? A Study of Tie formation and decay. *Advances in Strategic Management*, *23*, 481–519. https://doi.org/10.1017/CBO9781107415324.004

Siciliano, M. D. (2015). Advice Networks in Public Organizations: The Role of Structure, Internal Competition, and Individual Attributes. *Public Administration Review*, *75*(4), 548–559. https://doi.org/10.1111/puar.12362

Singh, P., & Tan, Y. (2010). Developer heterogeneity and formation of communication networks in open source software projects. *Journal of Management Information Systems*, *27*(3), 179–210. https://doi.org/10.2753/MIS0742-1222270307

Singh, P. V., Tan, Y., & Mookerjee, V. (2011). Network Effects: The Influence of Structural Capital on Open Source Project Success. *MIS Quartely*, *35*(4), 813–829. https://doi.org/10.1017/CBO9781107415324.004

Šmite, D., Moe, N. B., Šāblis, A., & Wohlin, C. (2017). Software teams and their knowledge networks in large-scale software development. *Information and Software Technology*, *86*, 71–86. https://doi.org/10.1016/j.infsof.2017.01.003

Smith-Doerr, L., Manev, I. M., & Rizova, P. (2004). The meaning of success: Network position and the social construction of project outcomes in an R&D lab. *Journal of Engineering and Technology Management - JET-M*, *21*(1–2), 51–81. https://doi.org/10.1016/j.jengtecman.2003.12.004

Smith, E. B., Menon, T., & Thompson, L. (2012). Status Differences in the Cognitive

Activation of Social Networks. *Organization Science*, *23*, 67–82.

Snijders, T. A. B., Lomi, A., & Jasmine, V. (2013). A model for the multiplex dynamics of two-mode and one-mode networks , with an application to employment preference, friendship, and advice. *Social Networks*, *35*, 265–276.

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, *104*, 333–339. https://doi.org/10.1016/j.jbusres.2019.07.039

Sowe, S., Stamelos, I., & Angelis, L. (2006). Identifying knowledge brokers that yield software engineering knowledge in OSS projects. *Information and Software Technology*, *48*(11), 1025–1033. https://doi.org/10.1016/j.infsof.2005.12.019

Stajkovic, A. D., Bandura, A., Locke, E. A., Lee, D., & Sergent, K. (2018). Test of three conceptual models of influence of the big five personality traits and self-efficacy on academic performance: A meta-analytic path-analysis. *Personality and Individual Differences*, *120*(November 2017), 238–245. https://doi.org/10.1016/j.paid.2017.08.014

Stenmark, D. (2002). Information vs. knowledge: The role of intranets in knowledge management. *Proceedings of the Annual Hawaii International Conference on System Sciences*, 928–937. https://doi.org/10.1109/HICSS.2002.994043

Storey, M. A., Zagalsky, A., Filho, F. F., Singer, L., & German, D. M. (2017). How Social and Communication Channels Shape and Challenge a Participatory Culture in Software Development. *IEEE Transactions on Software Engineering*, *43*(2), 185–204. https://doi.org/10.1109/TSE.2016.2584053

Strader, T. J., Frank, G. L., Houle, P. A., & Miller, C. E. (2008). The Impacat of Personality Type on Blog Participation. In *Proceedings of the Third Midwest United States Association for Information Systems, Eau Claire, WI May 23-24, 2008*.

Strode, D. E., Huff, S. L., Hope, B., & Link, S. (2012). Coordination in co-located agile software development projects. *Journal of Systems and Software*, *85*(6), 1222–1238. https://doi.org/10.1016/j.jss.2012.02.017

Su, C., & Contractor, N. (2011). A multidimensional Network Approach to Studying Team Members's Information Seeking From Human and Digital Knowledge

Sources in Consulting Firms. *Journal of The American Society for Information Science and Technology*, *62*(7), 1257–1275.

Subramaniam, N., Nandhakumar, J., & Baptista John, J. (2013). Exploring social network interactions in enterprise systems: The role of virtual co-presence. *Information Systems Journal*, *23*(6), 475–499. https://doi.org/10.1111/isj.12019

Sulayman, M., Urquhart, C., Mendes, E., & Seidel, S. (2012). Software process improvement success factors for small and medium Web companies: A qualitative study. *Information and Software Technology*, *54*, 479–500. https://doi.org/10.1016/j.infsof.2011.12.007

Sykes, T. A., Venkatesh, V., & Johnson, J. L. (2014). Enterprise System Implementation and Employee Job Performance: Understanding The Role of Advice Networks. *MIS Quarterly*, *38*(1).

Taillard, M., Peters, L. D., Pels, J., & Mele, C. (2016). The role of shared intentions in the emergence of service ecosystems. *Journal of Business Research*, *69*(8), 2972–2980. https://doi.org/10.1016/j.jbusres.2016.02.030

Tang, F. (2015). When communication quality is trustworthy? Transactive memory systems and the mediating role of trust in software development teams. *R and D Management*, *45*(1), 41–59. https://doi.org/10.1111/radm.12051

Tassabehji, R., Mishra, J. L., & Dominguez-Péry, C. (2019). Knowledge sharing for innovation performance improvement in micro/SMEs: an insight from the creative sector. *Production Planning and Control*, *30*(10–12), 935–950. https://doi.org/10.1080/09537287.2019.1582101

Tehseen, S., Sajilan, S., Gadar, K., & Ramayah, T. (2017). Assessing Cultural Orientation as a Reflective-Formative Second Order Construct-A Recent PLS-SEM Approach. *Review of Integrative Business and Economics Research*, *6*(2), 38. Retrieved from http://buscompress.com/journal-home.html

Tekinerdogan, B., & Aksit, M. (2006). Integrating the Concept of Synthesis in the Architecture Design Process. *Transactions of the SDPS*, *10*(1), 45–56.

Thomas, J. C., Lee, A., & Danis, C. (2002). Enhancing Creative Design via Software Tools. *Communications of the ACM*, *45*(10), 112–115. https://doi.org/10.1145/570907.570944

Tiwana, A. (2004). An empirical study of the effect of knowledge integration on software development performance. *Information and Software Technology*, *46*(13), 899–906. https://doi.org/10.1016/j.infsof.2004.03.006

Tiwana, A. (2012). Novelty-knowledge alignment: A theory of design convergence in systems development. *Journal of Management Information Systems*, *29*(1), 15–52. https://doi.org/10.2753/MIS0742-1222290101

Tuli, K. R., Bharadwaj, S. G., & Kohli, A. K. (2010). Ties That Bind: The Impact of Multiple Types of Ties with a Customer on Sales Growth and Sales Volatility. *Journal of Marketing Research*, *47*(1), 36–50. https://doi.org/10.1509/jmkr.47.1.36

Ulhas, K. R., Lai, J. Y., & Wang, J. (2016). Impacts of collaborative IS on software development project success in Indian software firms: a service perspective. *Information Systems and E-Business Management*, *14*(2), 315–336. https://doi.org/10.1007/s10257-015-0285-4

van Bel, D. ., Smolders, K. C. H. ., Ijsselsteijn, W. A., & de Kort, Y. (2009). *Social connectedness: Concept and measurement*. *Intelligent Environments*. IOS Press. https://doi.org/10.3233/978-1-60750-034-6-67

Vargas, N., & Schafer, M. H. (2013). Diversity in action: Interpersonal networks and the distribution of advice. *Social Science Research*, *42*(1), 46–58. https://doi.org/10.1016/j.ssresearch.2012.08.013

Venkatraman, N., & Lee, C. (2004). Preferential Linkage and Network Evolution : a Conceptual Model and Empirical Test in the U . S . *The Academy of Management Journal*, *47*(6), 876–892.

Verbeke, W., Dietz, B., & Verwaal, E. (2010). Drivers of sales performance: A contemporary meta-analysis. Have salespeople become knowledge brokers? *Journal of the Academy of Marketing Science*, *38*(3), 407–428. https://doi.org/10.1007/s11747-010-0211-8

Verbrugge, L. M. (1979). Multiplexity in Adult Friendships. *Social Forces*, *57*(4), 1286–1309.

Vermerris, A., Mocker, M., & Van Heck, E. (2014). No time to waste: The role of timing and complementarity of alignment practices in creating business value in IT projects. *European Journal of Information Systems*, *23*(6), 629–654.

https://doi.org/10.1057/ejis.2013.11

Verona, G., Prandelli, E., & Sawhney, M. (2006). Innovation and virtual environments: Towards virtual knowledge brokers. *Organization Studies*, *27*(6), 765–788. https://doi.org/10.1177/0170840606061073

vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R., & Cleven, A. (2015). Standing on the shoulders of giants: Challenges and recommendations of literature search in information systems research. *Communications of the Association for Information Systems*, *37*(9), 205–224. https://doi.org/10.17705/1cais.03709

Waizenegger, L., McKenna, B., Cai, W., & Bendz, T. (2020). An affordance perspective of team collaboration and enforced working from home during COVID-19. *European Journal of Information Systems*, *29*(4), 429–442. https://doi.org/10.1080/0960085X.2020.1800417

Wang, M.-H., Huang, C.-F., & Yang, T.-Y. (2012). The Effect of Project Environment on the Relationship between Knowledge Sharing and Team Creativity in the Software Development Context. *International Journal of Business & Information*, *7*(1), 59–80. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=91653684&site=ehost-live

Wang, Z., & Wang, N. (2012). Knowledge sharing, innovation and firm performance. *Expert Systems with Applications*, *39*(10), 8899–8908. https://doi.org/10.1016/j.eswa.2012.02.017

WeiQiang, Z., Li Ming, N., He, J., Zhen Yu, C., & Jia, L. (2014). Developer social networks in software engineering: construction, analysis, and applications. *Science China Information Sciences*, *57*(12), 1–23. https://doi.org/10.1007/s11432-014-5221-6

Wellman, B. (1983). Network Analysis: Some Basic Principles. *Sociological Theory*, *1*, 155–200. https://doi.org/10.2307/j.ctv5rf5rj.4

Wellman, B., Salaff, J., Dimitrova, D., Gulia, M., & Haythornthwaite, C. (1996). Computer Networks as Social Networks : Collaborative Work , Telework , and Virtual Community. *Annual Review of Sociology*, *22*, 213–238.

Whelan, E., Anderson, J., & Donnellan, B. (2015). How IT and the Rest of the Business Can Innovate Together. *Communications of the Association for Information Systems*, *36*, 261–272.

Whelan, E., Anderson, J., van den Hooff, B., & Donnellan, B. (2015). How IT and the Rest of the Business Can Innovate Together. *Communications of the Association for Information Systems*, *36*, 261–271. https://doi.org/10.17705/1cais.03614

Whelan, E., & Conboy, K. (2014). Understanding the role of advice networks in software development. *ECIS 2014 Proceedings - 22nd European Conference on Information Systems*, 1–9.

Whelan, E., Golden, W., & Donnellan, B. (2013). Digitising the R&D social network: Revisiting the technological gatekeeper. *Information Systems Journal*, *23*(3), 197–218. https://doi.org/10.1111/j.1365-2575.2011.00384.x

Whelan, E., Parise, S., De Valk, J., & Aalbers, R. (2011). Creating employee networks that deliver open innovation. *MIT Sloan Management Review*, *53*(1), 37–44.

Whelan, E., Teigland, R., Donnellan, B., & Golden, W. (2010). How internet technologies impact information flows in R&D: Reconsidering the technological gatekeeper. *R and D Management*, *40*(4), 400–413. https://doi.org/10.1111/j.1467-9310.2010.00610.x

Whelan, E., Teigland, R., Vaast, E., & Butler, B. (2016). Expanding the horizons of digital social networks: Mixing big trace datasets with qualitative approaches. *Information and Organization*, *26*(1–2), 1–12. https://doi.org/10.1016/j.infoandorg.2016.03.001

Williamson, G. M., & Clark, M. S. (1989). Providing Help and Desired Relationship Type as Determinants of Changes in Moods and Self-Evaluations. *Journal of Personality and Social Psychology*, *56*(5), 722–734. https://doi.org/10.1037/0022-3514.56.5.722

Winch, G. M., & Courtney, R. (2007). The organization of innovation brokers: An international review. *Technology Analysis and Strategic Management*, *19*(6), 747–763. https://doi.org/10.1080/09537320701711223

Wirth, J., Maier, C., Laumer, S., & Weitzel, T. (2021). Laziness as an explanation for the privacy paradox: a longitudinal empirical investigation. *Internet Research*, (2015). https://doi.org/10.1108/INTR-10-2019-0439

Wu, Y. J., Antone, B., Srinivas, A., DeChurch, L., & Contractor, N. (2021). Teamwork in the time of COVID-19: Creating, dissolving, and reactivating network ties in response to a crisis. *Journal of Applied Psychology*, *106*(10), 1483–1492. https://doi.org/10.1037/apl0000969

Xiao, Z., & Tsui, A. S. (2007). When Brokers May Not Work : The Cultural Contingency of Social Capital in Chinese High-Tech Firms. *Administrative Science Quarterly*, *52*(1), 1–31.

Xu, P., & Ramesh, B. (2008). Impact of knowledge support on the performance of software process tailoring. *Journal of Management Information Systems*, *25*(3), 277–314. https://doi.org/10.2753/MIS0742-1222250308

Xu, Y., Kim, H. W., & Kankanhalli, A. (2010). Task and social information seeking: Whom do we prefer and whom do we approach? *Journal of Management Information Systems*, *27*(3), 211–240. https://doi.org/10.2753/MIS0742-1222270308

Xuan, Q., Gharehyazie, M., Devanbu, P. T., & Filkov, V. (2012). Measuring the effect of social communications on individual working rhythms: A case study of open source software. *Proceedings of the 2012 ASE International Conference on Social Informatics, SocialInformatics 2012*, 78–85. https://doi.org/10.1109/SocialInformatics.2012.17

Yang, H. L., & Tang, J. H. (2004). Team structure and team performance in IS development: A social network perspective. *Information and Management*, *41*(3), 335–349. https://doi.org/10.1016/S0378-7206(03)00078-8

Yang, S. W., Soltis, S. M., Ross, J. R., & Labianca, G. (Joe). (2021). Dormant tie reactivation as an affiliative coping response to stressors during the COVID-19 crisis. *Journal of Applied Psychology*, *106*(4), 489–500. https://doi.org/10.1037/apl0000909

Yin, R. (1994). Discovering the Future of the Case Study Method in Evaluation Research. *Evaluation Practice*, *15*(3), 283–290.

Zachariadis, M., Scott, S., & Barrett, M. (2010). Working Paper Series evidence from the economics of IS innovations, 1–26.

Zand, D. E. (1972). Trust and Managerial Problem Solving. *Administrative Science Quarterly*, *3*(3), 229–239. https://doi.org/10.1177/105960117800300306

Zerbini, F., & Castaldo, S. (2007). Stay in or get out the Janus ? The maintenance of multiplex relationships between buyers and sellers. *Industrial Marketing Management*, *36*, 941–954. https://doi.org/10.1016/j.indmarman.2007.06.003

Zhang, W., Cheung, S. C., Chen, Z., Zhou, Y., & Luo, B. (2019). File-level socio-technical congruence and its relationship with bug proneness in OSS projects. *Journal of Systems and Software*, *156*, 21–40. https://doi.org/10.1016/j.jss.2019.05.030

Zhang, X., & Venkatesh, V. (2013). Explaining Employee Job Performance: The Role of Online and Offline Workplace Communication Networks. *MIS Quarterly, 37*(3), 695–722.

Zhang, X., Venkatesh, V., & Huang, B. (2008). Students interactions and course performance: Impacts of online and offline networks. *International Conference on Information Systems (ICIS)*, 215.

Zhu, B., & Watts, S. A. (2010). Visualization of network concepts: The impact of working memory capacity differences. *Information Systems Research*, *21*(2), 327–344. https://doi.org/10.1287/isre.1080.0215

Zhuge, H. (2003). Workflow- and agent-based cognitive flow management for distributed team Cooperation. *Information and Management*, *40*(5), 419–429. https://doi.org/10.1016/S0378-7206(02)00061-7

Appendices

List of Definitions of Multiplexity Reviewed in Chapter 2 (Article 1)

Authors/ Years	Definition: Multiplexity is …	Phenomenon Perspective
Gimeno & Woo (1996, p. 4-5)	*"...multiple instances of the interdependence of mixed types, such as when firms experience simultaneously vertical and horizontal interdependence…or horizontal and symbiotic interdependence…"*	Economic related *(mainly on cooperation and competition)*
Lazega & Pattison (1999, p. 82-83)	"*...multiplex dyadic exchange structures in which a pair of lawyers exchange different types of resource, suggesting some complementarity of resource dependence…various triadic configurations involving multiple resource ties…*"	Resources related (advice, friendship, goodwill) *(mainly on cooperation & collective action)*
Haythornthwaite (2001, p. 211)	"*...the way in which multiple types of interactions are accomplished through the variety of media available for communication.*"	Communication-related (face-to-face, mails, Webboard, etc.) *(mainly on collaboration)*
Talmud & Mesch (2006, p. 139)	"*… different dimensions a relationship contains and is high when individuals are connected in multiple activities and discussions…diversity of flows…*"	Communication-related (offline, online); Resource-lated (discussions, activities) *(mainly on quality collaboration)*
Ferriani et al., (2012, p. 9)	*"...multiplex ties, those that are configured by the simultaneous presence of one social and one economic tie between two actors. "*	Social & economic-related

		(mainly on quality collaboration)
Shipilov (2012, p. 215)	*"…different relationships that the firm maintains with a particular group of partners… or different groups of partners… "*	Relationship-related *(mainly on strategic cooperation)*
Cardillo et al., (2013, p. 1)	*"…as a graph made of diverse relationships (layers) between its constituents…"*	Graph-related (multilayer)
Maggioni et al., (2013, p. 186)	*"…the same actors may be simultaneously embedded in multiple collaboration networks and thus involved in relationships of a different nature."*	Relationship-related *(mainly on collaboration)*
Aalbers et al., (2014, p. 843)	*"... combining both formal and informal aspects in a relation between two individuals…"*	Relationship-related (formal, informal) *(mainly on ideation)*
Autry et al., (2014, p. 52-53)	*"… as a status where two or more formalized supply chain organisations are connected by multiple types of relational content… a status where there is a presence of multiple integrated supply chain processes…"*	Resources and process-related *(mainly on supply chain integration)*
Pavan, (2014, p. 448)	*"… a system shaped by the entwinement of different on- and offline relations..."*	Communication-related: online & offline relations *(mainly on collaboration and coordination for collective action)*
Shipilov et al., (2014, pp. 449-451)	*"…as the extent to which a focal entity (a person, a team, or an organisation) derives its meaning and its potential for action from relations of multiple kinds with other entities.*	Relationship-related (memberships, third parties)

	…the presence of multiple types of relationships between the same actors…membership in different social networks…"	*(mainly on strategic cooperation)*
Marineau et al., (2018, p. 595)	*" multiplex conflict relationships, or those infused with both relationship and task conflict..."*	Relationship-related (friendship/task) *(mainly on advice seeking decisions)*

Questionnaire Related to the Study Presented in Chapter 3 (Article 2)

Social Network Study

Q1 National University of Ireland, Galway

This social network survey aims to understand how software developers' source multiple types of advice or knowledge (e.g., technical, business and career-related support) from multiple sources. This data will be anonymised to ensure your privacy and the protection of your identities. Specifically, your data will be protected, and responses not revealed to management. It will take less than 5 minutes to complete this survey. If you need more details, please contact me (Matthew Ajimati, at m.ajimati1@nuigalway.ie).

For further information about me and our Business Information Systems Group, visit:http://nuig.lero.ie/research-team/
Thank you for your willing participation.

○ I consent to participation

○ I do not consent to participation

End of Block: The purpose of this study is to understand how software developers access multip

---

Start of Block: Block 2

Q2 1. What is your name?

____

Q3 2. What is your gender?

○ Male

○ Female

Q4 3. Which category below includes your age?

- ○ Under 21
- ○ 21-29
- ○ 30-39
- ○ 40-49
- ○ 50-59
- ○ Over 60

Q5 4. What is the highest level of education you have completed?

- ○ Professional Diploma
- ○ Bachelor degree
- ○ Master degree
- ○ Doctoral degree

Q6 5. How long have you been employed?

○ Less than one year

○ 1-4 years

○ 5-8 years

○ Over 9 years

Q14 6. Which work title below describes your position at HPE? Please select multiple options if applicable.

☐ Software Engineer

☐ Tech Lead

☐ Architect

☐ Scrum Master

End of Block: Block 2

Start of Block: Block 3

Please answer the following questions about your advice or knowledge interactions with your coworkers. Offline communication refers to face-to-face interaction. Online communication refers to the internet and web-based interaction via email, instant messaging, Skype, social media and other audio or video conference.

Q7 7. Please identify which work colleagues you discuss technical problems with at least once a week? Please leave the row blank if you do not interact with that person at all. Please feel free to provide additional names not listed below who you interact with, of others within the organisation. Technical problems refer to issues around understanding the use of technologies to develop and deploy software systems.

	I discuss technical problems with this person at least once a week...	What medium do you usually use to discuss technical problems with this person (select both options if applicable)?		This person is a member...(select one option)	
	Yes ()	Offline ()	Online ()	Within my team ()	Outside my team ()

Q8 8. Please identify which work colleagues you discuss **business problems with** at least once a week? Please leave the row blank if you do not interact with that person at all. Please feel free to provide additional names not listed below who you interact with, of others within the organisation. Business problems refer to issues around understanding user requirements, development processes and workflow practices within your work or domain environment.

	I discuss business problems with this	What medium do you usually use to discuss business problems with	This person is a member...(select one option)

	person at least once a week...	this person (select both options if applicable)?			
	Yes ()	Offline ()	Online ()	Within my team ()	Outside my team ()

Q13 10. Any further comments below on some benefits of specific advice or knowledge interactions with your colleagues?

____

____

End of Block: Block 3

Start of Block: Block 3

Survey Output

Insights generated from this social network survey will be used to make recommendations on how best to improve and optimise organisational work outcomes.

Q12 Thank you for your time and valuable insights. You are sincerely appreciated for your participation in this survey.

The Interview Guide for the Study Presented in Chapter 3 (Article 2)

Research Question

How does software developer multiplex interaction with advice in offline and online networks relate to performance?

(performance as dependent variable or outcome variable; offline, online, technical advice, friendship advice as independent or predictor variables)

Multiplex ties: overlap of offline contact and online contact; overlap of technical advice and friendship advice

Respondent Profile							
Date	Time	Place	Respondent Names	Position (s)	Years of Experience	Interviews Duration	Gender

Network Measures		
Section A		
Background Insights	**Interviewees Job Description**	**Tick question (s) asked**
	Could you describe the nature of your work?	
	Would you consider your work project-based and why?	
	Is this project individuals or teams are driven, or both? And why?	
	Would you say you participate in more than one project? And why?	
	Could you briefly explain your role or task in this work?	
	The nature and kinds of Work Role Advice	**Theme Focus 1**

	Based on your role, what types of advice are important to performing your role? Does this advice help you to solve work problems, and how? Does this advice help you to be approach your work in new ways, and how? ***Please note that advice can be referred to information and knowledge needed, as forms of recommendations and guidance to get your work done.***	
	Again, advice as information and knowledge and social support to perform work	
Section B (More specific questions)		
Theme Focus 2	**Different Role of Advice Sources in Advice Relations**	**Tick question (s) asked**
	Where and how do you get or access this advice from? Within a group or outside a group or both?	
	In your workplace relationship, would you say you tend to give and or get some certain advice from some people and not others? And why this difference?	
Friendship Connection	Would you say you give and or get some work-related or non-work-related advice with someone you socialise with, for example, going for dinner, movies and shopping together? How would you qualify for such a person? And why?	

Technical Connection	Would you say you give to and or get some advice related to software systems with people you socialise with or does not socialise with to get work done? And why?	
	Could you describe your medium of communicating with one another in your work role?	
	What sort of workplace relationship do you use any of these media for?	
Theme Focus 3	**The Relevance of Communication Channels in Advice Relations**	
Online Connection	Would you rather say that you give and or get some advice over the internet or web-based interaction (depending on the communicating channels)? And what type of advice – related to your work or not do you exchange through this?	
Offline Connection	Would you rather say that you give and or get some advice through face-to-face interaction (depending on the communicating channels)? And what type of advice – related to your work or not do you exchange through this?	
Online + Offline Connection	Do you consider giving and or getting some advice over both internet (or web-based) and face-to-face interaction matters to your work role, or rather one is more important than the other, especially for certain advice? If so, why?	
Theme Focus 4	**The Role of Time in Advice Relations**	

Introducing "TIME" factor in advice relations	Would you say ***time*** plays a big role where or which medium you choose to access advice, and whom you talk to – such as a particular workplace relationship to do your work?	
	Would you say time spent on accessing a contact for advice will affect future contact with the person for additional advice or not? And why? Or Are you willing to do a trade-off with time-based on the importance of advice itself to be accessed for immediate and future work and non-work-related needs?	
Theme Focus	**The outcome of Advice Application on Work and Non-work-related Tasks**	
Assessing Advice Outcomes	How would you describe the effect or uses of different advice on your ability to perform your work role?	
	Would you rather say some advice relate to your expectation versus project or organisational expectation? And why?	
	Please explain specifically how some advice have directly and indirectly affected your work and non-work-related expectations or motivations. How did you determine or evaluate this effect on you and your work?	

Pilot Interview Transcript for the Study Presented in Chapter 3 (Article 2)

Selected, Typical Comments by Respondents Type						
Interviewees	LD	PS	PH	JM	AM	ES
Position	Assistant software engineer	Senior software engineer	Technical lead/Domain expert	Software Engineer/System Integrator	Software Engineer	Senior software engineer
Project activity	So, what we do is we take in a list of requirements, we'll break down the requirements into user stories, then you take on the user story, you implement it, you do the test for it and you ensure that it is deployed to the testing environments and test passed to the testing environments, because the	I'm currently focused on project applications, which is sort of development and operational problems and trying to automate the development of applications into the web.	I make some of the more technical decisions around how we implement things from a development point of view. I also do some development within that role but at times it might just be designing solutions and other developers to actually implement the solution. And I'm also responsible for areas within the overall system right, such as	At the moment we're involved with integrating software that we have running in a WAN environment onto a Salesforce environment. So, a lot of what we've been doing is moving software to the cloud basically. I'm working on the X support portal so it's a portal that external users go to when they want to find information	The nature of my work, I work in an agile way, and we get user stories. So, user stories are telling us to write up this functionality and code and they have an acceptance criterion attached to them. So, I take one of those user stories and I do the coding, I do the testing and then I demo it to whoever usually my	We'll say customer-facing product. I'm working on an internal product that serves the development team. So, it's a kind of secondary work on the build system for the main product. So, it's a kind of a core product in the actual pipeline if you know what I

	methodology we follow is Agile.		particular domains. There's a domain called software domain which is providing binaries to customers right and as a domain architect within that area, my responsibilities are making sure that infrastructure is up to date and evolving, performant and that any new features in that area are kind of my responsibility as well to just disarm and design.	about X products or have issues and want to enter a support case or they want to look up what the latest drivers are or what the latest security bulletins are. So, and that exists at the moment and it's running on mainly X hardware and its X software developed so we're porting some of that to the Salesforce environment	project, my team and my project owner. And they check that the acceptance criteria have been met and they either accept it or they tell me to go back and do more work. Then there are other bits as well. I do, I keep an eye on production so if there's a problem, our software eventually will go into production and be used by the customers, so we have to watch what errors come up and deal with them.	mean as opposed to being a part of the actual product that's delivered. It's a development tool I suppose if you want to look at it that way.
Problem approach	If you had a problem or something you couldn't	Yeah, in general, it's good to go and present	It's we have a conversation, and we say, 'I had time to	So, without the internet, you'd probably be stuck, and	If I'm banging my head against the wall,	Yeah, in my work at the moment

	quite implement, you might ask a colleague, you know ‘I have this problem here. Could I get a fresh pair of eyes?’ And sometimes that’s all it requires, is somebody to have a look at it and say ‘Oh I’ve seen that before. This is the solution.’	the situation and that you will get different answers and opinions and you will come up with a solution. Yeah, well probably I would ask if the approach I’m taking is the right one. If I’m asking someone with experience.	look at this and I found A, B and C, what do you guys think?’. ‘I think B looks like a good way to approach this, what do you think?’, right so it’s not a hierarchical you know, based kind of organisation.	you’d end up having to ask some other, unless you’ve done something similar before but there was something new, yeah it would take you longer to do because you’d have to experiment more, try out different things and get more advice from other people which would take their time and then you’d arrive at a solution.	which does happen, I would ask my scrum master and I would call a meeting with maybe people, at that point if I still don’t understand, call a meeting with people from other countries like Germany, India. Like if I know there’s someone on the German site who wrote the code I’m looking at and understands, I might call a meeting with them and someone else and kind of get four heads together and discuss the issue.	I'm kind of, I'm on an island maybe if you know what I mean. I'm probably the domain expert. So, I might bounce it off a couple of people that I know, that I can rely on that have some experience we’ll say or would have generally good intuition on stuff.

Project membership	Yes, from Architects that define the requirements and have the domain knowledge, to the other Engineers that you've been working alongside and you've to coordinate … other Engineers on the team and obviously you'd be coordinating what you're working on with them, you'd be coordinating deliveries with them, you'd be and then also we work with other teams that would be delivered, either we're	I currently work with a team, but I think there is a traditional team that is like everyone working on the same, to try to achieve something. We don't really do that but it's more like interdisciplinary with different teams.	So, I think talking to others gives you multiple perspectives and I think all our work is some sort of problem-solving for the most part and all problems resolved to be a census of ideas and it's another resource for ideas and it's a different way of looking at solutions as well. So, it's that differentiation of perspective I suppose is probably one of the primary things you get from colleague conversation, peer meetings, whatever, you know. And this ideal format anyway is that I prefer this.	We're encouraged here to try and help people and be helpful so you wouldn't – but if you were being drawn into another project and it was going to take some time to answer a question you had to do some research, you might approach your project owner or scrum master and say, 'I would need some time to do this work that helps someone else, some other team', and they might say, 'Okay we'll put in some time slot for that or we'll let you do it. So, then it becomes official work rather than hidden	I also ask the other team members, depending on what the knowledge is. I notice, I know my team now and I know who knows what in the team so if I have one problem I might go to 'x' while if I have another problem, I might go to 'y'. The team is diverse, and they have different abilities and different knowledge so, likewise, people come to me for certain things like I go to them. So sometimes we have, we	I suppose people that I've worked with and that I know, be like-minded or, do you know like that would be, depending on the problem area I suppose.

	depending on what they're delivering or they're depending on what we're delivering.			underneath work.	have features from two or three projects within the team and I can, it's up to me to go on and say oh I know a bit about that, I'll take that user story but yeah, they all, they all connect to a project, one, two, three projects somewhere, yeah.	
Communication media	For domain knowledge, it's more got to do with face-to-face interaction. So, to get the technical knowledge it would be less one on one interactions.	I prefer the physical, my colleague that sits behind me knows lots. I can turn my chair and ask him! So, it's much easier to be there, it's sometimes easier to express or to show something on your screen to	It's definitely like, a few people in a room with a whiteboard, that's all I'd want. I would prefer that to be online. Well online is fine you know you can share stuff and show things as well, but I think nothing beats	The internet yeah or over – over messenger usually we talk, or we can have Skype calls. And if people are at home they join online and usually then you'd say to someone, 'I	Yeah, I try to talk face to face with them because it tends to work better. Then sometimes I might email, sometimes I'll Skype, sometimes I'll stand up and walk over and	Well, I suppose in my case a lot of them are local to me so it would be like grab a coffee, go for a chat, that kind of thing. So, fairly informal.

		someone just next to you, or whatever. Two seats far from you than someone else who is online somewhere else.	a whiteboard and a few people who know what they're talking about. Or willing to talk about something. Mostly face to face. Some online but mostly face to face.	must have a word with you about something because it's only 15 minutes. So, I want to have a word with you today about this and working on this, do you know anything about it? Or I have to talk about something that you're working on. I need to talk to you about that. So, we would do that as well.	talk to people.	
Business knowledge relations	For the domain knowledge, from experience Software Engineers would know a lot of it, because they're used to working in the domain. So, you'll get a lot of it from		I need to also know what the business plans are, and I need to know what high-level architectural plans are. And I even need to know to a degree the certain strategies the company is taking and strategies		By domain knowledge, I mean knowledge of the business I'm trying to programme. We have a project owner, and the project owner is the person who is kind of in	

	Software Developers, but then you also where there are new requirements coming in, maybe something we're not aware of, that's when you would go to more Architect level.		within our group and plans within that group. So, I'm a domain architect I would discuss with a solution architect about the overall solution and, 'I was thinking of doing this, does that fit okay within that? What do you think?'. So, that's probably the more inter-personal advice is talking to people around me, you know, in related roles.		charge of the project or projects. And they are not supposed to be technical and low down, they are supposed to be kind of at more the customer level, understanding what the customer needs. But domain knowledge is understanding the products we have, how they, you know, how they work together and how we are.	
Technical knowledge relations	Your technical knowledge would be more, it would allow you to be more	You basically need to know tools within those applications, for example, how to use the tools to get	Yeah, so I suppose what I need to have is an understanding of the current landscape or infrastructure.	So again, nearly all are Google, that's not to say we use it for everything because we have	So, there is the code I write, I need to know how to write the code, I need to know how	Yeah, it's all technical really. It's just knowing the technical

	innovative, more efficient, obviously greater quality in your code and in the code you deliver.	these applications deployed and tested and build those applications, to be able to put them in the final stage. So yeah, you need to know the tools, how the tools work	I also need to kind of be aware of current development methodologies and implementations and best practices and technologies being used and languages and so on. So, it's a broad enough range	sometimes the problem would be with our code or some code we've developed and we need to know maybe what's doing or can we change it and we might go to someone in our own company that worked on that previously or is an expert in that area, subject area experts we call them and we have one guy in our team who did a lot of work with Salesforce before the rest of us, he joined from outside, he's from Brazil and we often ask him you know, 'We're having a problem here you know did you come across this before?' And he might say, 'Yes or no',	to run the code, which is the environment, where do I run it, how do I set it up. I need to know the business logic behind the code. And then sometimes I need to know hardware bits and pieces as well, not as much but like if my laptop crashes, which it does on a regular basis. I need to also know knowledge of developer tools like there are different developer tools out there, tones of developments and you	domain, I think. It's trying to understand I think a broad technical domain that like as I said, especially with a third-party kind of tool or something like that. You haven't written it. So, we'll say if I was working on the development team, 90% of the stuff you'd be working with someone has written it like you can go and contact them or you can read the code may

				but he'd be able to probably tell you where to go…to look up this information.	have to learn them. And I would say going into soft skills, I need to know, that's all the technical side	be okay and figure out what is actually happening versus what you want to happen, or what you want to add onto it.
Personal and career support relations/goals	Maybe as you talk to people you get knowledge about career advancemen t. You would change your expectations maybe about promotion prospects and stuff like that.	In general, I just get better and more proficient in what I do and be able to achieve the goals with better solutions and generally try to, yeah, optimize the solutions to benefit my employer and get the knowledge.	So, it is primarily knowledge, but I mean, I suppose you can have more informal types of advice. That is one thing. That's solving the company's problems, but I would say, 'Listen, I'm thinking of going for a promotion, what do you think?'. I'm seeking advice from a peer or a manager or something like that which is about career development, not necessarily	Non-technical you know I'm thinking of things like pensions.	We have three things we're supposed to act on in here. You're supposed to innovate, act and partner. I gave them partner recognition because they worked with me well. Innovate is I design something, and act is I do something. So, they are across all the organisation	I find getting, like if you get technical training or something like paid training, that I think you have to be at a level of knowledge before you go in. Like I've often gone into training where I knew nothing, and you can't really ask the

			about solving a problem.		, they're called the values of the company. And my goals fit into those meaning I have to when I do my review.	right questions I suppose. I suppose the one thing is learning to ask the right questions. Like I think you need to kind of do a little bit of that hard work yourself to figure out and then you know, pick the brain and someone that is an expert in it or whatever to put you on the right path then after that.
Expected project output (*performance*)	A failed test would indicate that there's a problem. Defects being raised would	Yeah, I guess for the management it will be the time that it's taking you to do something and the time	Yeah, well usually you're making a design decision so how you're going to develop	We have monthly reviews with our manager who's not our actual work boss but is kind of a	Mmm, I, know they have a recognition system in here now where say we have	I suppose reliability of what you've produced if you know what I mean. So, I

	indicate poor quality. And badly performing software would indicate poor quality. Meeting requirements is probably our most important one. A very efficient software that doesn't meet your requirement is no good. Whereas if you have software that meets requirements, but has poor quality and efficiency, it's not very usable, but it has a bit of value. But if you have software that doesn't meet requirements, it has no value.	that it takes navigation from the source to the end-user. It can vary so that's something that management can see oh this navigation took two weeks to get out there and now it's taking three days, so they see an improvement in the release of an application.	something, a piece of software or a feature and I guess these people help with the process and the output of a design and the output of a design is an application or a feature.	people manager, but he gets feedback from the various people, I guess. It's individual but it's usually for example in our team we might decide someone gave us a good bit of help from another team and we might send an email to our scrum master to their manager, thanking them and giving them some recognition. Well, it would help at your review at the end of the year, they might be looking at salary increases, the manager might have a little leeway about how much salary increase you get.	reviews coming up for Christmas that will relate to our pay rises and our bonuses and stuff like that, sometimes the bonuses are cat – it depends on the profits. And I would say to do that, you must fill in saying I did this…, I did this right.	suppose fewer bugs. Then that means you can move on to something else, if you push something out with bad quality, you're going to end up going back to fix it constantly which means later work is going to be affected by the support of, you know, something with bad quality. .

Questionnaire Related to the Study Presented in Chapter 4 (Article 3)

End of Block: Block 3

Software Developer Advice Network Study

Start of Block: Default Question Block

Q7 National University of Ireland, Galway

This questionnaire is about how you experience your social relationship and social contacts with people in your business and technical advice networks. A business advice network relates to the network of people with information and knowledge needed to address business processes and customer needs such as analysing requirements. A technical advice network relates to the network of people with information and knowledge needed to solve specific technical problems to develop solutions such as writing and changing source codes to fix bugs. With contacts, we mean problem-solving conversations/communications in each others' company or media (telephone, chat, email, text messaging, etc) towards developing software products.

This data will be anonymised to ensure your privacy and the protection of your identities. It will take less than 5 minutes to complete this survey.

If you need more details, please contact me (Matthew Ajimati, at m.ajimati1@nuigalway.ie). Further information about me and our Business Information

Systems Group, visit:http://nuig.lero.ie/research-team/

Thank you for your willing participation.

Q22 Your Prolific ID

____

Q9 Q1. What year were you born?

____

Q9 Q2. What is your gender?

○ Male

○ Female

○ Non-binary, third

○ Prefer not to say

Q10 Q3. What is the highest level of education you have completed?

○ Professional diploma

○ Bachelor degree

○ Master degree

○ Doctorate degree

Q14 Q4. Which of the following best describes your educational background?

○ Engineering

○ Humanities

○ Health sciences

○ Natural sciences

○ Social sciences

Q11 Q5. How long have you been employed by your company?

____

Q12 Q6. Which of the following best describes your position in this company?

○ Junior software developer

○ Senior software developer

○ Junior, non-software developer

○ Senior, non-software developer

Q15 Q7. How much computer experience do you have in years?

○ Less than one year

○ 1-4 years

○ 5-8 years

○ over 9 years

Q13 Q8. Which of the following best describes your ethnic background?

- ○ Whites
- ○ Blacks
- ○ Asians
- ○ Hispanics
- ○ Prefer not to say

Q16 Q9. Which of the following best describes your communication in location access to work colleagues over the past two months?

- ☐ Face-to-face office meetings
- ☐ Virtual office meetings
- ☐ Face-to-face/Virtual meetings

Q3. Q10. Please reflect on the statements below about your experience with people within your ***business advice network*** in providing you with information and knowledge related to problem-solving to get your work done. A *business advice network* relates to the network of people with information and knowledge needed to address business and customer needs such as analysing requirements [1=completely disagree, 7=completely agree].

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. I often know what people in my business advice network think	○	○	○	○	○	○	○
2. I often know what people in my business advice network feel	○	○	○	○	○	○	○

3. I am often aware of my relationships with people in my business advice network	○	○	○	○	○	○	○

Q19 Q11. Please rate the extent to which the statements below reflect your ***social orientation*** in connecting other work colleagues within your ***business advice network***...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. I introduce people to each other who might have a common strategic work interest	○	○	○	○	○	○	○
2. I will try to describe an issue in a way that will appeal to a diverse set of interests	○	○	○	○	○	○	○

3. I see opportunities for collaboration between people	○	○	○	○	○	○	○
4. I point out the common ground shared by people who have different perspectives on an issue	○	○	○	○	○	○	○
5. I introduce two people when I think they might benefit from becoming acquainted	○	○	○	○	○	○	○

6. I forge connections between different people dealing with a particular issue	○	○	○	○	○	○	○

Q25 12. Please rate the extent to which the statements below reflect ***the quality of your social interactions*** within your ***business*** *advice network*...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. The social contacts with people in my business advice network feel superficial	○	○	○	○	○	○	○
2. My relationships with people in my business advice network feel superficial	○	○	○	○	○	○	○

3. I derive little satisfactio n from my social contacts	○	○	○	○	○	○	○

Q8 Q13. Please reflect on the statements below about your experience with people within your ***technical advice network*** in providing you with information and knowledge related to problem-solving to get your work done. A *technical advice network* relates to the network of people with information and knowledge needed to solve specific technical problems to develop solutions such as writing and changing source codes to fix bugs [1=completely disagree, 7=completely agree].

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1.I often know what people in my technical advice network think	○	○	○	○	○	○	○
2. I often know what people in my technical advice network feel	○	○	○	○	○	○	○

3. I am often aware of my relationships with people in my technical advice network	○	○	○	○	○	○	○

Q20 Q14. Please rate the extent to which the statements below reflect your *social orientation* in connecting other work colleagues within your ***technical*** *advice network*...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. I introduce people to each other who might have a common strategic work interest	○	○	○	○	○	○	○
2. I will try to describe an issue in a way that will appeal to a diverse set of interests	○	○	○	○	○	○	○

3. I see opportunities for collaboration between people	○	○	○	○	○	○	○
4. I point out the common ground shared by people who have different perspectives on an issue	○	○	○	○	○	○	○
5. I introduce two people when I think they might benefit from becoming acquainted	○	○	○	○	○	○	○

6. I forge connections between different people dealing with a particular issue	○	○	○	○	○	○	○

Q24 15. Please rate the extent to which the statements below reflect ***the quality of your social interactions*** within your ***technical*** *advice network*...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. The social contacts with people in my technical advice network feel superficial	○	○	○	○	○	○	○
2. My relationships with people in my technical advice network feel superficial ()	○	○	○	○	○	○	○

3. I derive little satisfaction from my social contacts ()	○	○	○	○	○	○	○

Q4. Q16. Please rate the extent to which the statements below reflect how your ***social relationship*** and ***social contacts*** in the advice networks help you...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. Quickly define, find, and implement solutions to the problems	○	○	○	○	○	○	○
2. Evaluate a large number of alternative solutions for each problem quickly	○	○	○	○	○	○	○

3. Found and implemented the right number of alternative solutions to problems	○	○	○	○	○	○	○
4. For each problem encountered with spoiled choice of solutions	○	○	○	○	○	○	○
5. Implemented solutions found in less cost than in previous projects	○	○	○	○	○	○	○

6. Came up with lower-cost solutions for problems than expected	○	○	○	○	○	○	○
7. Found and Implemented creative solutions	○	○	○	○	○	○	○
8. Generate fresh thinking from the solutions found	○	○	○	○	○	○	○

Q5. Q17. Please reflect on your preference for ***online versus offline face-to-face communication*** with your colleagues [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. I prefer online social interaction over face-to-face communication	○	○	○	○	○	○	○
2. Online social interaction is more comfortable for me than face-to-face interaction	○	○	○	○	○	○	○
3. I prefer communication with people online rather than face-to-face	○	○	○	○	○	○	○

Q20 Q18. Please rate the extent to which the statements below reflect your behaviour: ***I could complete a job or task using a technology system...*** [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. If there was no one around to tell me what to do as I go.	○	○	○	○	○	○	○
2. If I could call someone for help if I got stuck.	○	○	○	○	○	○	○

3. If I had a lot of time to complete the job for which the software was provided.	○	○	○	○	○	○	○
4. If I had just the built-in help facility for assistance.	○	○	○	○	○	○	○

Q17 Q19. Please rate the extent to which the statements below reflect your ***approach to work*** with colleagues...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. My peers often ask me for advice or information	○	○	○	○	○	○	○
2. I enjoy trying new ideas	○	○	○	○	○	○	○
3. I seek new ways to do things	○	○	○	○	○	○	○
4. I am generally cautions about accepting new ideas	○	○	○	○	○	○	○

5. I consider myself to be creative and original in my thinking and behaviour	○	○	○	○	○	○	○

Q18 Q20. Please rate the extent to which the statements below reflect your ***work demand and interaction with colleagues***...? [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. Quality of work	○	○	○	○	○	○	○
2. Quantity of work	○	○	○	○	○	○	○
3. Technical competence	○	○	○	○	○	○	○
4. Working as part of a team or work group	○	○	○	○	○	○	○
5. Help others when it is not part of his/her job	○	○	○	○	○	○	○

Q21 Q21. Please rate the extent to which the statements below reflect your ***self-awareness***: ***I ...*** [1=completely disagree, 7=completely agree]

	1=Completely disagree ()	Disagree ()	Somewhat disagree ()	Neutral ()	Somewhat agree ()	Agree ()	7=Completely agree ()
1. am always prepared	○	○	○	○	○	○	○
2. pay attention to details	○	○	○	○	○	○	○
3. make plans and stick to them	○	○	○	○	○	○	○
4. waste my time.	○	○	○	○	○	○	○
5. find it difficult to get down to work.	○	○	○	○	○	○	○

Q7 *Thank you for your time and active participation in this survey. For follow up, do indicate "Yes" or "No" below if you could be contacted for a further interview or feedback opportunities as the survey closes. If yes, please leave your email in the box below. You can also contact me via email:* *m.ajimati1@nuigalway.ie* *for further information.*

○ Yes (4) ____

○ No (5)

End of Block: Default Question Block

Measurement Scales Related to the Study Presented in Chapter 4 (Article 3)

Business Advice Connectedness (BAC)
BAC1. I often know what people in my business advice network think
BAC2. I often know what people in my business advice network feel
BAC3. I am often aware of my relationships with people in my business advice network
Brokerage in Business Advice Connectedness (BBAC)
BBAC1. I introduce people to each other who might have a common strategic work interest
BBAC2. I will try to describe an issue in a way that will appeal to a diverse set of interests
BBAC3. I see opportunities for collaboration between people
BBAC4. I point out the common ground shared by people who have different perspectives on an issue
BBAC5. I introduce two people when I think they might benefit from becoming acquainted
BBAC6. I forge connections between different people dealing with a particular issue*
Contact Quality in Business Advice Connectedness (CQBAC)
CQBAC1. The social contacts with people in my business advice network feel superficial
CQBAC2. My relationships with people in my business advice network feel superficial

CQBAC3. I derive little satisfaction from my social contacts
Technical Advice Network (TAC)
TAC1.I often know what people in my technical advice network think
TAC2. I often know what people in my technical advice network feel
TAC3. I am often aware of my relationships with people in my technical advice network
Brokerage in Technical Advice Connectedness (BTAC)
BTAC1. I introduce people to each other who might have a common strategic work interest
BTAC2. I will try to describe an issue in a way that will appeal to a diverse set of interests
BTAC3. I see opportunities for collaboration between people
BTAC4. I point out the common ground shared by people who have different perspectives on an issue
BTAC5. I introduce two people when I think they might benefit from becoming acquainted
BTAC6. I forge connections between different people dealing with a particular issue*
Contact Quality in Technical Advice Connectedness (CQTAC)
CQTAC1. The social contacts with people in my technical advice network feel superficial
CQTAC2. My relationships with people in my technical advice network feel superficial
CQTAC3. I derive little satisfaction from my social contacts
Problem-Solving Competence (PSC)
PSC1. Quickly define, find, and implement solutions to the problems
PSC2. Evaluate a large number of alternative solutions for each problem quickly
PSC3. Found and implemented the right number of alternative solutions to problems
PSC4. For each problem encountered with spoiled choice of solutions*
PSC5. Implemented solutions found in less cost than in previous projects*
PSC6. Came up with lower-cost solutions for problems than expected*
PSC7. Found and implemented creative solutions
PSC8. Generate fresh thinking from the solutions found
Conscientiousness Personality (CP)
CP1. I am always prepared
CP2. I pay attention to details
CP3. I make plans and stick to them*
CP4. I waste my time*
CP5.I find it difficult to get down to work*
Computer Self-Efficacy (CSE)
CSE1. If there was no one around to tell me what to do as I go*
CSE2. If I could call someone for help if I got stuck
CSE3. If I had a lot of time to complete the job for which the software was provided.

** Items removed from PLS-SEM regression model due to low factor loadings (<0.70)*

Research Ethics and Data Management

In accordance with the ethics of research conducts set up by both NUI Galway Research Ethics Committee (see http://www.nuigalway.ie/media/staffsubsites/researchoffice/files/10.pdf and Radboud Ethics Committee (see https://www.ru.nl/nsm/imr/about-imr/research-integrity/ethics-committee/ethics-self-assessment-criteria/, accessed 25 October 2018); the following protocols were adhered to.

The data gathered involved semi-structured interviews with software developers. To avoid any confidentiality concerns about some questions relating to their exchange of information and knowledge with particular individuals during the interviews, I ensured that a written confidentiality agreement document was signed, to maintain privacy during the process of interviews and in survey questionnaires. The interviews were conducted in transparent manners and within the premises of the case organisation.

In a strict sense, I ensured that the interviews and survey questionnaires participation was voluntary for the empirical studies. I also ensured that the anonymity of participants was considered during the analysis and reporting of accessible information and in published work. This was done by coding the names of participants and the participating firm to maintain privacy and data protection. Based on the consent of the interviewees, the interviews were recorded, transcribed, and stored in NVivo software analytic tool for the cross-sectional study conducted in this thesis. The audio interviews recorded was transcribed into text readable in the English language for use, reuse and sharing. They were further stored in the personal computer database and allocated space of U: Drive allocated 100Mb of disk space allowable for each student in the university storage server, where they can be easily accessed. NUI Galway Research Ethics Committee permits data retention for 3 years after which it will be destroyed.

More importantly, the used and stored data will be destroyed later on when it is no longer needed. As the right to this collected data belongs to the research investigator, further usage will need permission from the researcher. In case the information could be useful for related research purposes by other NUI Galway and Radboud university researchers, there shall be interviewees informed consent before access is granted to the data. Finally, appropriate keywords will be suggested for retrieving specific contents of this stored data across storage spaces.

Sex and Gender Dimension

This thesis considered whether gender perceptions might affect the result of this study. Gender bias has been a topical but polarised issue of concern. Despite the large proportion of women versus men in our society, women are mostly underrepresented in occupying a prestigious position and in key decision-making capacity and performance reward across the social, political, economic, and religious spectrum. Given that this research focused on advice interactions (information and knowledge exchange) of software developers, the gender dimension was taken into consideration. While this thesis considered gender distribution of samples among the representative demographics of the studied population, this thesis revealed a lack of gender bias in the empirical findings or results of this study.