

## EQUALITY DOES NOT MAKE YOU HAPPY: EFFECTS OF DIFFERENTIATED LEADER-MEMBER EXCHANGE AND TEAM-MEMBER EXCHANGE ON DEVELOPER SATISFACTION IN AGILE DEVELOPMENT TEAMS<sup>1</sup>

**Viswanath Venkatesh**

Department of Business Information Technology, Pamplin College of Business, Virginia Tech, Blacksburg, VA, U.S.A. {vvenkatesh@vvenkatesh.us}

**James Y. L. Thong**

Department of ISOM, School of Business and Management, Hong Kong University of Science and Technology, HONG KONG {jthong@ust.hk}

**Kai Spohrer**

Department of Management, Frankfurt School of Finance & Management, Frankfurt am Main, GERMANY {k.spohrer@fs.de}

**Frank K. Y. Chan**

ESSEC Business School, Cergy Pontoise Cedex, FRANCE {chanf@essec.edu}

**Ankur Arora**

Department of Business Information and Technology, University of Memphis, Memphis, TN, U.S.A. {ankur.arora@memphis.edu}

**Hartmut Hoehle**

Department of Enterprise Systems, Business School, University of Mannheim, Mannheim, GERMANY {hoehle@uni-mannheim.de}

**Srinivasan Venkatraman**

The Boeing Company, Charleston, SC, U.S.A. {mail.srini.raman@gmail.com}

---

*Prior work on leadership in information systems development (ISD) teams has assumed that all developers are treated equally by their team leader and ignored the possibility that differentiated leader-member exchange (LMX) may be an important instrument for team leaders to influence self-organizing, agile ISD teams. We conducted a concurrent mixed methods inquiry to understand how LMX differentiation is associated with developer satisfaction in agile ISD teams and through which team processes agile ISD teams address LMX differentiation. We ran a multilevel, multistage survey of 1,894 software developers in 217 teams and an embedded case study of five ISD teams drawing on qualitative data from 40 interviews of developers and team leaders. Two focus groups (one with 10 developers and one with 10 team leaders) helped to substantiate the meta-inferences from the quantitative and qualitative studies. The results showed that LMX differentiation was positively associated with developer satisfaction, especially in teams with high-quality team-member exchange (TMX). We identify three team processes (i.e., collectivization of resources, visible appreciation of privileges, and freeing up leader capacities) that are enacted through agile ISD practices and allow ISD teams to leverage benefits from LMX differentiation for all their members.*

**Keywords:** Leader-member exchange differentiation, LMX differentiation, team-member exchange, TMX, agile software development, developer satisfaction, social exchange theory, team processes, mixed methods

---

<sup>1</sup> Saonee Sarker was the accepting senior editor for this paper. Narayan Ramasubbu served as the associate editor. This paper was accepted on November 21, 2022, when Saonee Sarker was employed at Lund University.



## Introduction

How leaders of information systems development (ISD) teams can beneficially influence their developers is a key topic in information systems (IS) research and practice (PMI, 2017; Venkatesh et al., 2018; Windeler et al., 2017). While ISD team leaders engage in important control activities to achieve beneficial project outcomes (Maruping et al., 2009; Maruping et al., 2019; Venkatesh et al., 2018), it is similarly important that they effectively manage their interpersonal relationships with their developers. In fact, interpersonal relationships with team leaders are regularly seen as the top criterion influencing developer satisfaction (Johnson et al., 2021; Storey et al., 2019). Given that developers' job and team satisfaction—defined as the extent to which a developer has a positive affective reaction toward his/her job and team, respectively—drive organizational citizenship behavior and performance, reduce absenteeism, and prevent turnover (Johnson et al., 2021; Tripp et al., 2016), it is vital that ISD team leaders manage interpersonal relationships in a way that benefits developer job and team satisfaction.

Managing interpersonal relationships has become even more challenging for ISD team leaders with the adoption of agile ISD methods (e.g., extreme programming—XP and Scrum) because agile ISD methods require new leadership behaviors (Shastri et al., 2021a). Instead of exerting formal control and hierarchical task assignment, agile ISD team leaders need to facilitate self-organization and collaboration among team members through intensive coaching and feedback provision (Maruping et al., 2009; Windeler et al., 2017). Despite the importance of leadership in ISD (PMI, 2017), however, little is known about how ISD team leaders should manage interpersonal relationships with their team members, leaving them with open questions and challenges (Shastri et al., 2021a, 2021b). For example, coaching and supporting senior team members to their satisfaction may require more effort and different means than coaching new graduates. Given that team leaders' time and resources are limited, intensive coaching and support for one team member may come at the expense of coaching and support for others. Thus, ISD team leaders need guidance on how to simultaneously manage their interpersonal relationships with all team members.

The ISD leadership literature suggests that, besides individual developer's organizational skills (Venkatesh et al., 2020), specific leadership styles and leader control behaviors can reduce developer stress and work exhaustion, increase developer performance, and steer peer pressure toward beneficial outcomes (Khanagha et al., 2022; Venkatesh et al., 2018; Windeler et al., 2017). However, these studies have all assumed that leader behaviors are equally directed toward all

team members. In doing so, they have overlooked that differentiated treatment of team members may be an important tool for ISD team leaders to influence developers. By contrast, the leadership literature in other disciplines has established that leaders build personal relationships with individual employees that range “from low-quality transactional relationships based on formal contractual exchanges (low leader-member exchange (LMX)) to high-quality socio-emotional relationships that supplement the formal contractual exchange with mutual trust, loyalty, obligation, and commitment (high LMX)” (Matta & van Dyne, 2020, p. 154).

Importantly, team leaders' interpersonal relationships with one team member are not independent of their relationships with other team members. In fact, a growing stream of research on LMX differentiation—defined as the variability in LMX relationships within a team (Erdogan & Bauer, 2010)—has noted that dyadic interpersonal relationships between team leaders and members are embedded within the social structure of a work group and can affect its members, processes, and performance (Martin et al., 2018; Matta & van Dyne, 2020). Yet findings on the effects of LMX differentiation are highly inconsistent (see Henderson et al., 2009; Martin et al., 2018; Yu et al., 2018). For example, Hooper and Martin (2008) found LMX differentiation to stimulate intrateam conflict and reduce team members' job satisfaction, whereas others found that LMX differentiation can facilitate intrateam relations and increase job satisfaction (e.g., Epitropaki et al., 2016; Erdogan & Bauer, 2010). There are compelling explanations for both potential beneficial and detrimental effects of LMX differentiation on team members (Matta & van Dyne, 2020). On the one hand, team members may perceive LMX differentiation as beneficial because it allows leaders to reward and direct resources to those members who use them most effectively for the good of the team; on the other hand, team members may perceive LMX differentiation as discriminatory because some members do not receive the resources they desire or need. In line with this ambiguity, reviews and meta-analyses have concluded that unknown team-level factors and processes could play a key role in how LMX differentiation affects teams and their members (Anand et al., 2014; Martin et al., 2018; Yu et al., 2018). It is, however, not clear why and how some teams benefit from LMX differentiation, whereas others do not.

Against this backdrop, it is important to understand how LMX differentiation influences developers in self-organizing, agile ISD teams. Given that software companies recognize developers as their most valuable assets and invest heavily in improving developer satisfaction (e.g., industry surveys<sup>2</sup>; Johnson et al., 2021), we particularly need to understand how LMX differentiation influences developers' satisfaction

<sup>2</sup> <https://insights.stackoverflow.com/survey>; <https://about.gitlab.com/developer-survey>; <https://igda.org/dss/>

with their jobs and their teams. In this work, we examine the possibility that existing social exchange relationships between team members (i.e., team-member exchange, TMX) and specific team processes may help ISD team members benefit individually and collectively from LMX differentiation. Collaborative ISD practices, such as the ones entailed in agile ISD methods, may fuel team processes that leverage the benefits of LMX differentiation. Thus, we engage in context theorizing and aim to understand context-specific team processes in agile ISD teams (Hong et al., 2014) by answering two research questions (RQ):

1. *How are LMX differentiation and TMX in agile ISD teams associated with developer satisfaction?*
2. *How do team processes mitigate the impact of LMX differentiation in agile ISD teams?*

We used a concurrent mixed methods research design, with completeness as the primary purpose<sup>3</sup> (Venkatesh et al., 2013; Venkatesh et al., 2016). We addressed RQ1 through a multilevel, multistage survey of 217 agile ISD teams, with a total of 1,894 developers to examine the associations of LMX differentiation and TMX with developer job satisfaction and developer team satisfaction. We addressed RQ2 by conducting an embedded case study of five ISD teams via 40 interviews with developers and team leaders. We integrated the findings from the two studies into meta-inferences and corroborated them with two focus groups. Our research contributes to the IS leadership literature by explaining how LMX differentiation interacts with TMX in agile ISD teams, benefits developers, and gets reinforced through three team processes.

## Theoretical Foundations

### *Leadership in Agile ISD Teams*

ISD team leaders play an essential role in supporting their developers and in facilitating processes related to interpersonal relationships (Windeler et al., 2017). A recent survey of ISD teams found that developers perceive their manager to have the strongest influence on their job satisfaction and productivity compared to all other factors (Storey et al., 2019). However, the authors emphasized that the exact mechanisms underlying these effects are unclear and that little is known about how ISD team leaders should

manage their personal relationships with team members. Storey et al. (2019, p. 14) specifically highlighted that “consideration should be given to how managers show appreciation, give feedback on work, and how they promote a positive team and work culture.”

The popularity of agile ISD methods has reinforced a trend toward servant leaders who aim to facilitate self-organization and collaboration in ISD teams rather than formally directing and controlling ISD work in a top-down manner (Maruping et al., 2009; PMI, 2017). Agile ISD methods emphasize self-organized teams and short, incremental development iterations with many informal and personal interactions during which developers jointly make sense of unexpected developments, come up with solutions, and adapt to change (Chan & Thong, 2009; Ghobadi & Mathiassen, 2017; Hong et al., 2011; Sarker & Sarker, 2009). Accordingly, prior leadership research on ISD teams has shown that effective leaders stimulate interpersonal processes between team members so that they can jointly achieve team goals. Venkatesh et al. (2018) suggested that leaders can mitigate ISD risks through interpersonal control that reduces developer stress and improves developer performance; Maruping et al. (2009) found that team autonomy is most effective when leaders complement it with outcome control and promote teamwork; and Khanagha et al. (2022) suggested that leaders’ control behaviors can steer peer pressure in agile ISD teams toward or away from beneficial outcomes. Some studies have shown that empowering leadership, which provides significant decision-making autonomy and support to teams, has beneficial effects on developers, such as improved role perceptions, self-reliance, proactivity, knowledge sharing with team members, and performance in uncertain tasks (Chen et al., 2019; Faraj & Sambamurthy, 2006; Windeler et al., 2017).

A key takeaway from prior work is that ISD team leaders have a strong influence on both the inner workings and outcome achievement of their teams, regardless of whether the teams apply agile ISD methods or not. However, this line of research does not speak specifically to the dyadic relationships between team leaders and team members or how these dyadic relationships are embedded in the overall social structure of a team. Prior work on leadership in ISD teams has assumed that all developers are treated equally by their team leader while ignoring the possibility that differentiated treatment of individuals may be an important instrument for team leaders to influence ISD teams.

<sup>3</sup> In mixed methods research, a completeness purpose is when we seek to gain a more comprehensive picture of the phenomenon to include a rich

understanding of the mechanisms (Venkatesh et al., 2013; Venkatesh et al., 2016).

## **LMX Differentiation**

Prior work on LMX suggests that team leaders engage in LMX to make subordinates act in line with the leader's goals, within as well as beyond the scope of their role descriptions (Anand et al., 2010). In forming close relationships with subordinates, leaders can increase subordinates' job satisfaction (Cropanzano & Mitchell, 2005). In return, subordinates are more likely to act in the interest of their leader, thus reducing the need for formal control (Graen & Uhl-Bien, 1995). Although LMX is a significant determinant of positive individual work attitudes (Li & Liao, 2014), the interdependencies between a leader's multiple LMX relationships have been understudied in LMX research (Yu et al., 2018). Given that it is costly to establish and maintain high-quality interpersonal relationships, "leaders differentiate their exchange quality with their subordinates to make the most effective use of their limited resources, such that more resources are provided to subordinates who can contribute more toward accomplishing collective objectives" (Matta & van Dyne, 2020, p. 157).

Previous findings on the effects of LMX differentiation on team members are mixed and often contradictory (see Anand et al., 2014; Martin et al., 2018; Yu et al., 2018). Both positive and negative associations of LMX differentiation with team performance, group emergent states, individual performance, and individual satisfaction have been identified (Anand et al., 2014; Martin et al., 2018; Yu et al., 2018). The crucial question of how low-LMX team members react to the high-LMX relationships of their colleagues has not been sufficiently investigated. Some studies suggest that perceptions of justice and fairness determine how team members react to LMX differentiation (e.g., Erdogan & Bauer, 2010; Matta & van Dyne, 2020), others suggest that specific LMX distribution patterns may lend insights (Martin et al., 2018), and still others suggest that different kinds of teams rely on different team processes to cope with LMX differentiation, which then influences team members' reactions (e.g., Anand et al., 2014). The latter perspective is supported by meta-analyses showing that positive associations of LMX differentiation with group performance are more often observed in product development teams than in other professions and industries (Yu et al., 2018). Accordingly, scholars lament that we know too little about context-dependent group processes through which leader-member relationships influence outcomes in work groups (e.g., Anand et al., 2014; Farh et al., 2017; Yu et al., 2018).

In sum, there are two important gaps in understanding LMX differentiation and leadership in ISD teams. First, prior research on leadership in ISD teams has assumed that leadership styles and control behaviors are applied equally to all developers. It does not provide guidance on whether

and how ISD team leaders can engage in social exchange with individual developers without creating unwanted side effects on other team members. Second, prior work on LMX differentiation has not fully accounted for the role of team processes through which social exchange influences outcomes in teams. Initial evidence suggests that some ISD teams benefit from LMX differentiation and most other professions rarely do (Yu et al., 2018). We thus need to uncover and understand the processes by which ISD teams respond to LMX differentiation.

## **Social Exchange Theory**

Although different theoretical lenses can be used to study LMX and LMX differentiation (Yu et al., 2018, p. 1160), we use social exchange theory (SET), given its focus on social exchange between team leaders and team members as well as between team members (Banks et al., 2014). SET holds that social exchange relationships cause mutual obligations between exchange partners and that individuals are motivated by the goal of benefiting from exchanged resources and the resulting obligations (Blau, 1964). In contrast to economic transactions, social exchange relationships do not rely on clearly defined terms of exchange but carry the notion that a resource given now will be reciprocated at some point in the future in a way that is not yet clearly defined (Blau, 1964). For example, individuals in organizations exchange economic resources, such as goods, contacts, influence, and information, as well as socioemotional resources, such as loyalty, affect, and attention (Cropanzano & Mitchell, 2005). On the one hand, social exchange relationships enable individuals to access resources that would otherwise be out of their reach (Anand et al., 2010). On the other hand, high-quality social exchange relationships make individuals perceive obligations toward their exchange partners (Cropanzano & Mitchell, 2005), put high pressure on them to reciprocate received benefits (Banks et al., 2014), and urge them to apply received resources (Farh et al., 2017). In fact, the refusal to accept and make use of resources is often as detrimental to a social exchange relationship as not repaying received favors (Blau, 1964, p. 107). From an SET perspective, ISD team leaders establish high LMX so that the privileged developers feel obliged to act in line with the leaders' goals. Through high LMX, team leaders provide privileged team members with access to valuable resources that are hard to access for other employees. The privileged team members then feel obliged to reciprocate received benefits and are likely to act in the interest of the leader, aiming to reinforce the relationship. Social exchange relationships between individual developers within a team follow the same mechanisms of resource exchange, obligation, and reciprocation.

## Theory Development

### *Developer Satisfaction and LMX Differentiation in Agile ISD Teams*

The core thesis of our research model on LMX differentiation in agile ISD teams (Figure 1) is that teams rely on specific team processes to transform benefits received by a few team members (those with high LMX) into collective benefits for all team members, thus increasing team members' satisfaction with their jobs and their team. We reason that these effects are conditioned by social exchange relationships within the team (i.e., TMX).

Agile ISD methods entail iterative work processes, advocate self-organization, and prescribe regular interactions within the ISD team for collaboration and knowledge sharing (Ramasubbu & Bardhan, 2021; Venkatesh et al., 2018). In agile ISD, team leaders must engage less in formal control and more in actively managing interpersonal relationships, coaching, and mentoring (Maruping et al., 2009). At the same time, they are responsible for facilitating the team's work by removing external obstacles, securing necessary resources, and coordinating with external entities (Shastri et al., 2021, 2021a). Yet agile ISD team leaders face similar constraints in time, energy, and resources as leaders of other types of teams and thus cannot establish high LMX with all their team members (Matta & van Dyne, 2020). Consequently, they may establish high LMX with few selected team members, providing these selected team members with particularly high-quality resources, and low LMX with others (Graen & Uhl-Bien, 1995). For example, agile ISD team leaders may provide insights into the organization's strategy, expensive training in cutting-edge technology, intensive mentoring, and contact with upper management primarily to those developers who are most crucial for the team's success. At the same time, team leaders may withdraw some personal attention from routine challenges and issues faced by less crucial or low-performing developers and leave them to the self-organization of the agile ISD team.

Establishing high LMX with the developers who are most important for the overall team goals ensures that these developers receive the resources they need to be successful. Thus, LMX differentiation can be expected to drive their job satisfaction. However, LMX differentiation also affects those developers who cannot rely on high LMX but observe that others can. Following SET, such team members tend to become eager to leverage alternative exchange relationships—for example, the ones with the rest of their team—to compensate for the resource inequalities they observe (Anand et al., 2010, 2014; Farh et al., 2017; Herdman et al., 2017). If they succeed in compensating for the resource inequalities over time, their job satisfaction will also increase.

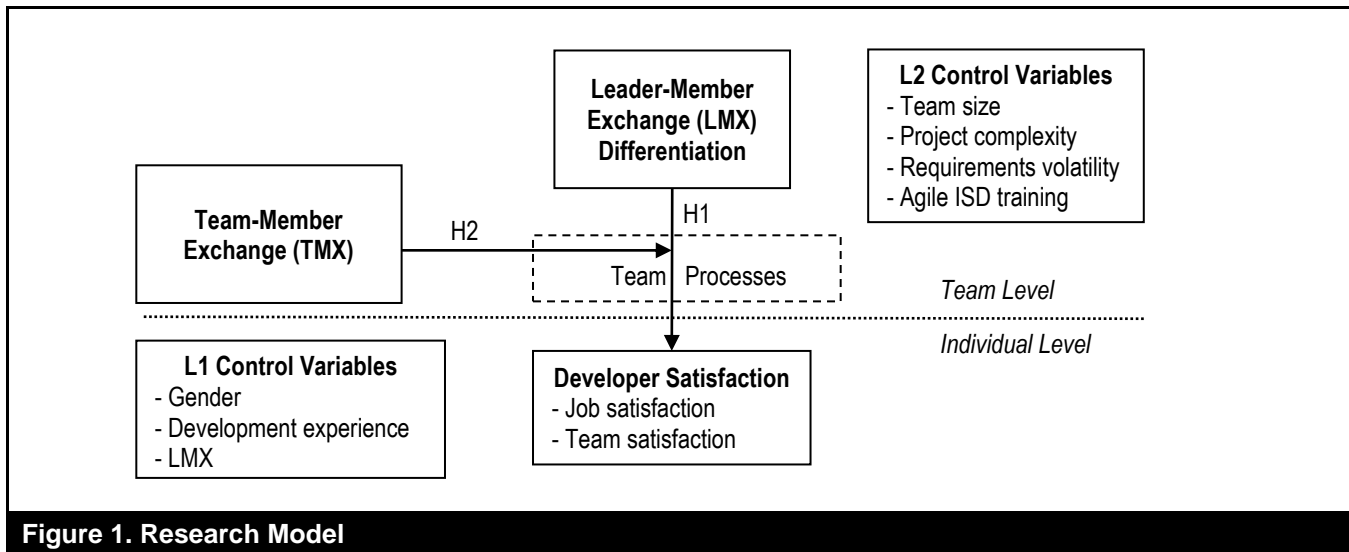
Given that agile ISD methods entail practices and values that promote joint problem solving and knowledge sharing (Chan & Thong, 2009; Ghobadi & Mathiassen, 2017; Hong et al., 2011; Ramasubbu & Bardhan, 2021), members of agile ISD teams have the means at hand to compensate for leader-induced resource inequalities. Low-LMX developers may engage in collaborative agile practices with their high-LMX colleagues in order to benefit from leader-provided resources that they cannot access directly. High-LMX developers may agree to such collaborations as they feel obliged to reciprocate privileges received from the leader by directing their team's actions toward the leader's goals (Herdman et al., 2017; Yu et al., 2018). In agile ISD, these goals necessarily include self-organization, collaboration, and knowledge sharing. For example, a low-LMX developer may engage in pair programming with a high-LMX developer who has received training in a cutting-edge technology in order to learn how the technology can be used. High-LMX developers may engage in refactoring and dialogs with low-performing team members to educate them on coding standards and quality expectations. Thus, routine challenges and issues of low-performing developers, from whom the team leader withdraws resources, become opportunities for high-LMX developers to demonstrate their contribution to the self-organization of the team.

Overall, unequally distributed, high-quality resources may stimulate both high-LMX and low-LMX team members to engage in agile practices in order to access, apply, and share resources. Ultimately, LMX differentiation should result in higher job satisfaction for high-LMX developers (as they can directly draw on high-quality social exchange with their leader) and low-LMX developers (as they can indirectly access the resources from which they would otherwise be cut off). Importantly, if team processes are effectively used to access, apply, and share high-quality resources in line with overall team goals, then LMX differentiation should not only leave developers more satisfied with their own resources but also increase developer team satisfaction. Thus, we hypothesize:

**H1:** *LMX differentiation is positively related to (a) developer job satisfaction and (b) developer team satisfaction.*

### **TMX as a Moderator**

Social exchange is not restricted to relationships between a team leader and team members but is also common between individual team members (Farmer et al., 2015). TMX refers to the overall quality of the exchange relationship between the members of a work group and the work group as a whole (Seers et al., 1995).



**Figure 1. Research Model**

In teams with high-quality TMX relationships, members identify with the team, feel strong mutual obligations to act in each other's interest, and have a general tendency to refrain from opportunistic behavior at the cost of their teammates (Farmer et al., 2015). Consequently, each member can count on the others' willingness to reciprocate beneficial actions directly or indirectly and becomes more willing to exert nonmandatory efforts for the benefit of the group and its members (Seers et al., 1995). Members in teams with high-quality TMX relationships are more likely to share resources, such as information and ideas, and aim to align their personal perspectives in order to achieve beneficial outcomes with regard to team goals (Farmer et al., 2015).

We expect that TMX amplifies the positive relationships of LMX differentiation with developer satisfaction. We theorize that high-LMX developers engage in resource exchange with their low-LMX team members in order to reciprocate the leader's favors. In teams with high-quality TMX relationships, privileged developers have an additional incentive to apply and share their high-quality resources in team tasks. Developers in high-quality TMX relationships can assume that their team members will reciprocate their efforts. Given that high-LMX developers can share particularly valuable resources during collaborative agile ISD practices, they can create particularly high obligations on the part of their team members (Farmer et al., 2015). Conversely, low-LMX developers can more easily tolerate LMX differentiation if they know that high-LMX developers will share their resources for the benefit of the team. In high-quality TMX relationships, agile practices and collaboration with high-LMX colleagues provide an effective way for low-LMX developers to access and leverage the privileged team members' resources. In teams with low-quality TMX

relationships, by contrast, high-LMX developers may feel less compelled to interact with their colleagues because they assume that their efforts and the valuable resources they bring in will not be reciprocated. Thus, we hypothesize:

**H2:** *TMX moderates the relationship of LMX differentiation with (a) developer job satisfaction and (b) developer team satisfaction, such that the positive relationship is stronger for higher levels of TMX.*

## Method and Results

We conducted a concurrent mixed methods investigation of ISD teams in one software company, consisting of a quantitative survey and an embedded case study. The primary purpose of our mixed methods study was completeness (see Venkatesh et al., 2013; Venkatesh et al., 2016). The quantitative study (survey) allowed us to test the research model of associations of LMX differentiation and TMX with developer satisfaction, and rich data from the qualitative study (embedded case study) allowed us to unearth previously unknown team processes (mechanisms) that underlie these associations.

Both data collections were conducted concurrently, following Venkatesh et al. (2013, p. 38): "If the broad goal of an IS research inquiry is to understand a phenomenon as it happens (e.g., a new IS implementation, a software development project), a concurrent mixed methods design approach should be employed." We used a nested sample (Venkatesh et al., 2016) by randomly choosing five teams that participated in the survey for the qualitative study to ensure tight alignment of

our two data collections. Survey data of the five teams helped to verify our subjective impressions and qualitative assessments of LMX differentiation and TMX in these teams (we did not find any mismatches between the survey and case data), and case study data helped to rule out that participants felt there was a major confounding context variable that we had not captured in the survey (at least for the five case study teams).

After completing the analyses of both studies, we derived meta-inferences from the sum of their results (Venkatesh et al., 2013). Whereas our survey results allowed us to test the deductively hypothesized associations of LMX differentiation and TMX with developer satisfaction, our case study results allowed us to abductively elicit the team processes. The integration of such a set of results that relate to different objects is well-suited to providing a more complete picture of the domain under study (Venkatesh et al., 2016, p. 449). In integrating the results, we moved abductively back and forth between SET as a theoretical frame and the two studies, aiming to make sense of them and establish a logical connection between them (Venkatesh et al., 2016, p. 448). As is typical for such a process (Sarker et al., 2018, p. 759), our preconceptions and creative reasoning played a substantial part in this process. To ensure the high quality of our meta-inferences, we conducted a concise empirical corroboration with two focus groups, although most mixed methods studies settle for assessing meta-inference quality only based on the quality of inferences from the underlying studies (Venkatesh et al., 2013). In Appendix A, we show how we conformed to quality criteria for mixed methods research designs.

We conducted the study in a large Indian software company. The company had chosen XP as its focal ISD method, trained its employees in XP, and encouraged the use of XP practices in ISD teams. Hence, all developers in our sample had a basic understanding of agile ISD methods and XP in particular. In line with agile principles (Ramasubbu & Bardhan, 2021; Sarker & Sarker, 2009), the ISD teams had the freedom to decide how to best accomplish their work. The company encouraged team leaders to decide how to best manage the relationships with their team members, provide them with feedback, and coach them. As is generally the case, different team leaders in our study chose to exert different degrees of LMX differentiation. Whereas some software vendors enforce the use of specific ISD methods or even single ISD practices top-down (Bick et al., 2018; Ramasubbu et al., 2015), this company did not impose such constraints. Instead, the ISD teams had the autonomy to decide on the ISD method and practices that they used and how they applied them. Consequently, the ISD teams' actions and interactions provided a good basis for understanding if and how agile practices were used to reap the benefits of LMX differentiation.

## Survey Study

### Data Collection

Survey data were collected at individual and team levels in three stages from different sources during ISD projects. The projects lasted between 80 and 140 workdays and focused primarily on the custom development of software for client companies. In a first stage survey at the start of the projects, we measured TMX and LMX. In a second stage survey, when the projects were well underway but not yet winding down, we again measured TMX and LMX (correlations of these variables across points of measurement were  $>0.70$ ) and control variables related to agile ISD, namely agile ISD training and development experience. At the end of the projects, we measured key project characteristics (e.g., project complexity) and two outcome variables (i.e., developer job satisfaction and developer team satisfaction).

The sampling frame of our survey consisted of 3,989 software developers who worked in ISD teams. The company's top management invited their developers to participate in our survey. We discarded responses from those who did not complete all the surveys. We restricted our sample to projects that had no overlap in terms of developers to prevent confounding effects of multiple project/team membership. Our final sample consisted of individual-level data from 1,894 developers (58% men) and team-level data on the 217 ISD projects on which they worked in the course of the year (we focused on projects that could be studied from start to finish).

The average age in our sample was 29.4 years ( $SD = 4.75$ ) for team members and 41.66 years ( $SD = 3.81$ ) for team leaders. The average team member's tenure at the company was 2.98 years ( $SD = 1.60$ ); average team leader's tenure was 6.97 years ( $SD = 1.49$ ). All developers had experience in software development ( $M = 3.1$  years,  $SD = 1.8$ ) and agile ISD methods ( $M = 1.9$  years,  $SD = 1.5$ ). To examine potential nonresponse bias, we obtained demographic data of those in the sampling frame and compared them to our sample. Comparing respondents and nonrespondents, there were no significant differences in terms of gender ( $p = 0.24$ ), age ( $p = 0.28$ ), tenure ( $p = 0.16$ ), development experience ( $p = 0.08$ ), or experience in XP ( $p = 0.14$ ).

### Items and Control Variables

We used previously validated scales for all variables and adapted them, where necessary, to fit the research context (Appendix B). We measured developer team satisfaction with the scale by Windeler et al. (2017) and job satisfaction based on Morris and Venkatesh (2010). TMX was measured with 10

items developed by Seers et al. (1995). LMX was measured with the LMX-7 scale (Graen & Uhl-Bien, 1995). Consistent with prior work (Anand et al., 2014), we calculated LMX differentiation within a team as the standard deviation of LMX-7 scores reported by members of the team.

We controlled for team size because larger teams tend to have higher coordination costs that may impact their social interactions, performance, and satisfaction, and leaders of larger teams face more challenges if they want to develop high(er) LMX with many team members. As developers' emotional states can be significantly related to project characteristics and how well the developers are equipped to cope with these characteristics (Venkatesh et al., 2018), we controlled for project complexity using the scale by Rai et al. (2009), requirements volatility as perceived by team leaders using the scale from Maruping et al. (2009), and teams' agile ISD training as perceived by the team members using the scale from Igbaria et al. (1997) adapted to training in XP. Regarding the cross-level effects of LMX differentiation on developers, we controlled for developer gender and development experience because they may affect the outcomes of agile ISD (Tripp et al., 2016). Finally, we controlled for each developer's LMX because we were interested in the effects of LMX differentiation rather than the direct effects of LMX (Banks et al., 2014). TMX and agile ISD training were the only team-level variables that were calculated as the mean of responses of developers on each team. All other team-level control variables were measured directly at the team level.

## Data Analysis

To test our nested research model, we used random coefficient modeling (RCM), which is regarded as a sophisticated technique that accounts for the non-independence of nesting arrangements (Mathieu & Chen, 2011). Prior studies on LMX differentiation (e.g., Anand et al., 2010; Erdogan & Bauer, 2010; Li & Liao, 2014) and leadership in ISD (e.g., Venkatesh et al., 2018; Windeler et al., 2017) have used RCM. Thus, analyses based on RCM allow an equitable comparison to prior work (see Appendix C for details on our RCM approach).

Table 1 presents the descriptive statistics and correlations. Appendix C provides further details on variable distributions. All scales displayed good reliability, convergent validity, and discriminant validity. The internal consistency reliability (ICRs) and Cronbach's alpha exceeded 0.70 (where applicable), thus indicating adequate reliability. We assessed convergent and discriminant validity using a factor analysis with oblimin rotation allowing for correlated factors. All items

loaded significantly on their specified constructs, with loadings greater than 0.70 and cross-loadings below 0.35. For all factors, the average variance extracted (AVE) was greater than 0.50. The highest heterotrait-monotrait ratio (HTMT) of correlations in our model was 0.44 and thus comfortably lower than the conservative threshold of 0.85. The square root of the AVE was greater than the interconstruct correlations for all factors, supporting discriminant validity. The team-level aggregation of TMX and agile ISD training from developer responses was supported by substantial variance attributable to between-team differences ( $ICC(1)_{\text{Agile ISD training}} = 0.37$ ;  $ICC(1)_{\text{TMX}} = 0.38$ ), reliability of team-level means greater than 0.70 ( $ICC(2)_{\text{Agile ISD training}} = 0.80$ ;  $ICC(2)_{\text{TMX}} = 0.72$ ), and average  $r_{\text{wg}(j)}$  greater than 0.70 ( $r_{\text{wg}(j)}_{\text{Agile ISD training}} = 0.77$ ;  $r_{\text{wg}(j)}_{\text{TMX}} = 0.79$ ).

## Findings

Two two-level NULL models indicated sufficient variability on both team and developer levels in the RCM analysis. The NULL models attributed 15.8% ( $\chi^2 = 2,113.08$ ,  $p < 0.001$ ) of variance in developer team satisfaction and 18.5% ( $\chi^2 = 2,808.15$ ,  $p < 0.001$ ) of variance in developer job satisfaction to differences between teams. Table 2 presents the results of the two-level RCM estimates of the relationships between TMX and LMX differentiation at the team level and developers' satisfaction with their teams and their jobs at the individual level. Appendix C provides a discussion of the significance of control variables in the different models.

With regard to our hypotheses, LMX differentiation had significant positive associations with developers' satisfaction with their teams (Model 4,  $\gamma = 0.17$ ,  $p < 0.01$ ) and their jobs (Model 9,  $\gamma = 0.14$ ,  $p < 0.05$ ), thus providing support for H1a and H1b. We also found significant positive associations of TMX with developers' satisfaction with their teams (Model 4,  $\gamma = 0.28$ ,  $p < 0.001$ ) and their jobs (Model 9,  $\gamma = 0.26$ ,  $p < 0.001$ ). The interaction terms of TMX and LMX differentiation had significant associations with developer team satisfaction (Model 5,  $\gamma = 0.14$ ,  $p < 0.05$ ) and developer job satisfaction (Model 10,  $\gamma = 0.19$ ,  $p < 0.001$ ). We plotted the interactions one *SD* above and below the mean of TMX (see Figure 2). LMX differentiation and developer team/job satisfaction had a stronger positive relationship when TMX was high than when TMX was low. This lends support to H2a and H2b. The moderated two-level models explained up to 28% of variance in developer team satisfaction (Model 5) and up to 27% in developer job satisfaction (Model 10), proportionally reduced for Level 1 and Level 2 errors, per Snijders and Bosker (1999). Overall, the results provide strong support for our research model.



**Table 1. Descriptive Statistics and Correlations**

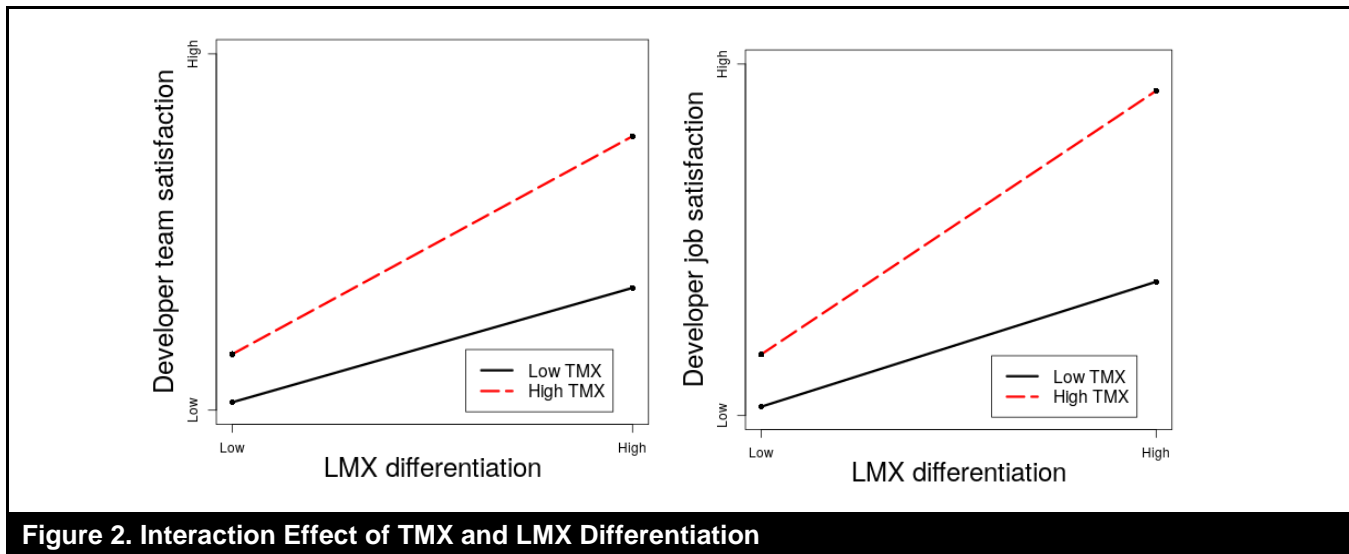
	Mean	SD	Alpha	1	2	3	4	5	6	7	8	9	10	11
1. Gender	.42	.50	NA	NA										
2. Development experience	3.1	1.80	NA	.19**	NA									
3. Team size	8.73	1.93	NA	.07	.05	NA								
4. Project complexity	.612	.114	NA	.14*	.26***	.23***	NA							
5. Requirements volatility	4.91	1.77	.75	.17**	.17**	.17**	.30***	.73						
6. Agile ISD training	4.56	1.46	.76	.14*	.15*	.05	.13*	.06	.74					
7. LMX	4.37	1.80	.78	.16**	.12*	-.10	-.05	-.12*	.13*	.74				
8. TMX	4.48	1.73	.83	.21***	.06	.12*	.24***	.07	.19**	.17**	.77			
9. LMX differentiation	3.12	1.66	.75	.13*	.20**	.14*	.21***	.09	.06	-.14*	.24***	.79		
10. Developer team satisfaction	5.51	1.38	.84	.16**	.24***	-.13*	-.21***	-.25***	.20**	.15*	.34***	.30***	.79	
11. Developer job satisfaction	5.13	1.81	.77	.15*	.05	-.08	-.10	-.08	.04	.19**	.23**	.13*	.26**	.73

**Note:** N = 1,894 developers (217 teams). Square root of AVE on the diagonal; Gender coded with female = 1. Gender, development experience, LMX, developer team satisfaction, and developer job satisfaction on Level 1 (individual developer); all other variables on Level 2 (ISD team). \* p < 0.05; \*\* p < 0.01; \*\*\* p < 0.001.

**Table 2. RCM Analysis Predicting Relationships with Developer Satisfaction**

Models	Developer team satisfaction				Developer job satisfaction					
	1	2	3	4	5	6	7	8	9	10
	<b>L1 control variables</b>									
Intercept	.08 (.052)	.06 (.033)	.08 (.044)	.07 (.038)	.09* (.041)	.02 (.044)	.13 (.071)	.15* (.063)	.10 (.072)	.12 (.083)
Gender	.14* (.059)	.12* (.048)	.13* (.053)	.13* (.055)	.12* (.054)	.13* (.059)	.08 (.097)	.08 (.041)	.07 (.030)	.06 (.031)
Development experience	.16** (.055)	.12* (.048)	.13* (.052)	.13* (.052)	.13* (.051)	.16** (.053)	.12* (.050)	.16** (.050)	.14* (.055)	.11* (.048)
LMX	.14* (.057)	.13* (.051)	.11* (.047)	.10* (.045)	.10 (.14)	.15* (.064)	.13* (.051)	.15* (.061)	.13* (.057)	.11* (.053)
	<b>L2 control variables</b>									
Team size	-.05 (.033)	-.04 (.037)	-.08 (.045)	-.07 (.043)	-.05 (.051)	-.16** (.05)	-.13* (.052)	-.14* (.058)	-.13* (.055)	-.11* (.052)
Project complexity	-.13* (.053)	-.11* (.044)	-.09* (.039)	-.08 (.051)	-.11* (.048)	-.17** (.051)	-.14* (.059)	-.14* (.054)	-.13* (.055)	-.08 (.051)
Requirements volatility	-.14* (.066)	-.14* (.055)	-.12* (.048)	-.12* (.051)	-.13* (.058)	-.13* (.053)	-.11* (.055)	-.04 (.033)	-.06 (.051)	-.08 (.044)
Agile ISD training	.15* (.059)	.12* (.047)	.13* (.057)	.13* (.053)	.12* (.053)	.10 (.073)	.07 (.065)	.09 (.024)	.09 (.048)	.10 (.080)
	<b>Team level exchange relationships</b>									
TMX		.30*** (.023)		.28*** (.025)	.25*** (.029)		.29*** (.030)		.26*** (.031)	
LMX differentiation			.19*** (.019)	.17** (.051)	.15* (.058)			.16** (.048)	.14* (.057)	.07 (.053)
TMXxLMX differentiation					.14* (.059)					.19*** (.017)
AIC	910.40	868.62	818.94	818.66	818.37	950.15	862.80	869.44	852.25	810.41
BIC	999.11	901.44	922.63	889.45	866.84	1,022.10	964.13	980.69	943.28	913.69
R <sup>2</sup>	.19	.23	.22	.25	.28	.17	.23	.23	.25	.27

**Note:** N = 1,894 developers (217 teams). \* p < 0.05; \*\* p < 0.01; \*\*\* p < 0.001. Robust standard errors in parentheses.



Following Venkatesh et al. (2018), we conducted multiple post hoc tests to minimize the threat of common method bias (CMB). We ran Harman's one-factor test (Podsakoff et al., 2003) with an unrotated factor analysis. The first factor extracted only about 12% of the variance, thus reducing CMB concerns. A marker variable test (Lindell & Whitney, 2001) resulted in an attenuation below 0.05 in correlations and the significance levels remained stable, thus further reducing concerns about potential CMB. Appendix D provides additional analyses regarding potential endogeneity issues. Specifically, we reduced potential concerns of reverse causality through a multistage survey design; we addressed potential concerns of selection bias through two-stage Heckman procedures with the leader's age as an instrumental variable; we assessed potential omitted variable bias by calculating the impact threshold of an omitted variable. None of the analyses suggested that endogeneity was a concern.

## Embedded Case Study

### Data Collection

Case study data from five ISD teams were collected via 40 interviews (five interviews with team leaders, 35 with developers). Appendix E provides background information on the teams. The interviews drilled down on how the interviewees perceived interpersonal relationships in their teams and how they used agile practices throughout their workdays. Interviews with team leaders focused on their attitude toward LMX differentiation, their relationships with different team members, and how they perceived the daily work of their teams. Primarily through open-ended questions, we aimed to unearth the mechanisms that these ISD teams leveraged to cope with LMX differentiation. Each

interview lasted around 30 minutes. We took field notes of responses and selected quotes during the interviews, which were archived and analyzed. Detailed field notes are a valuable data source (Miles & Huberman, 1994) and are commonly used in mixed methods research (e.g., Wunderlich et al., 2019).

### Data Analysis

Our qualitative data analysis drew on an initial set of codes that were derived from our literature review (Miles & Huberman, 1994, p. 58; Saldaña, 2013, p. 100). The predefined codes covered team members' TMX, LMX, satisfaction, and use of agile practices. Predefined codes helped us categorize the teams' TMX and LMX differentiation (Appendix E, Table E2). Next, one author started open coding statements about interactions of developers that were related to their leader or to managing resources that the leader provided or failed to provide. The authors discussed the codes and refined them to reach common interpretations. We proceeded by versus-coding (Saldaña, 2013, p. 115) the statements of high-LMX team members against those of low-LMX team members and compared the emergent patterns, aiming to make sense of how developers reacted to LMX differentiation in their development projects. Subsequently, we contrasted the patterns that emerged from teams with low and medium TMX to those that emerged from teams with high TMX. In a process of abduction (Sarker et al., 2018), we jumped between this bottom-up sensemaking and the guiding ideas of SET, namely the importance of resource exchange, obligation, and reciprocity.

We arrived at three mechanisms through which the agile ISD teams successfully leveraged advantages of LMX differentiation for the benefit of their members: (1) collectivization of resources, (2) visible appreciation of privileges, and (3) freeing up leaders' capacities. These mechanisms were embedded in the ISD teams' agile practices and were dependent on relationships between team members. Appendix E provides examples of predefined and open codes and illustrates the emergence of the three mechanisms.

## Findings

Although the team leaders enacted varying amounts of LMX differentiation in their teams, many team leaders saw LMX differentiation as a common and necessary way of allocating resources effectively. One team leader perceived it as follows:

*I mean, you cannot treat everybody the same. You need to provide everybody with what is best for them individually and for the team as a whole. Some are fresh from university, and you can basically send them to the standard trainings ... But then, my most important team member, ... if he comes to me to ask for resources, that can mean I have to give something exceptional that is not provided to everybody and I have and will do that.*

In line with the survey results, developers who experienced low LMX in their teams often viewed LMX differentiation critically, especially if they were not the ones who had close relationships with the team leader. Developers with low LMX described privileged colleagues with high LMX by using expressions, such as their "leader's buddies" who received "prestigious tasks" to work on, as opposed to the "menial debugging and hot-fixing" tasks on which they themselves worked. Some developers felt that their leaders saw low-LMX team members more as "a necessary evil" than as equally valuable parts of the team.

In contrast, teams with high-quality LMX leveraged the benefits of LMX differentiation. We found three mechanisms, enacted in agile practices, through which ISD leaders and teams jointly benefited from LMX differentiation (Table 3). First, teams with high-quality LMX relationships and high LMX differentiation used agile practices to collectivize the high-quality resources that a few privileged developers received from their leader. Privileged team members put the resources to work during daily interactions with their nonprivileged team members. Several teams reported that developers with good relationships with their leader were sent for training in cutting-edge technologies and to cross-departmental task forces more frequently than developers with low LMX. Privileged developers thereby acquired valuable

resources in the form of expert knowledge that was unique in their team. Once they returned from training or a task force, these privileged developers would engage with their team members to transfer knowledge about what they had learned. The developers suggested that several agile practices were useful for distributing such resources within a team. Some mentioned pair programming as a great opportunity to discuss and showcase how programming tasks could be accomplished using a new, cutting-edge technology or relying on extant functionality that had been developed in other departments. Others emphasized that collective code ownership and continuous integration were essential to distributing new knowledge about technology and cross-departmental activities in a team because these practices forced all team members to view, work on, and discuss the iterative improvement of their software with the newly gained hands-on knowledge. Overall, the collectivization of resources allowed team members with low LMX to access valuable resources that would otherwise have been inaccessible to them. The collectivization of resources thereby reinforced existing obligations and feelings of reciprocity between privileged and nonprivileged developers. One developer perceived it as follows:

*And then there are my team members [Member A] and [Member B]. They are much closer with our manager [than I am]; they can basically ask her any favor and she will try to make it happen. So, when I really have trouble and need something from our manager, I can always ask [Member A] or [Member B] to get it for me.*

Second, privileged developers with high LMX used agile practices to show visible appreciation to their team leader for the benefits they received. Several developers provided examples referring to meetings, such as planning game meetings and meetings with customers, in which privileged developers used information that only they had received from their leader to set the team in strategically important directions or convince customers of points that were important to the team leader. In stand-up meetings, privileged developers reportedly volunteered for boring but important tasks and openly contributed knowledge that they had gained through their privileges to design discussions. In applying their resources observably toward team goals, privileged developers reinforced their high LMX by reciprocating favors and advantages they received from their leader. One team leader's view of a typical situation is as follows:

*That's another reason why it is so important to have at least one team member you can trust. In my team, my senior architect is usually the one who comes to me and alerts me when he realizes that the team or some team members are stuck with a problem.*

**Table 3. Mechanisms for Leveraging Benefits of LMX Differentiation**

Mechanism	Focal relationship	Description	Exemplary agile practices	Function in social exchange
Collectivization of resources	Developers with high LMX and developers with low LMX	Privileged developers share the resources they receive from the leader with their colleagues during team task completion. Agile practices help team members access, share, and apply resources.	Pair programming, collective code ownership, continuous integration	Reinforce existing obligations between privileged and nonprivileged team members, create new obligations for team members, increase expectations of reciprocity within the team.
Visible appreciation of privileges	Developers with high LMX and team leader	Developers with high-quality LMX apply received resources for the good of the team when it is observable by the team leader.	Planning game, stand-up meetings, customer involvement	By visibly applying received resources, privileged developers show appreciation to the team leader and reinforce their high-quality LMX relationship.
Freeing up leader capacities	Developers with high LMX and team leader	Privileged developers participate in self-organization of the team and steer the team toward the leader's goals. Privileged developers make the team handle simple management tasks internally without leader involvement.	Iteration planning, refactoring, pair programming, coding standards	Privileged developers create further obligations for the team leader that reinforce their high-LMX relationship. Their engagement in self-organization of the team frees up leader capacities that can be used to acquire higher quality resources for the team.

Third, agile practices were used by privileged developers to free up leader capacities and thereby reciprocate received benefits. Engaging actively in the self-organization of their teams, developers with high LMX steered their team in the direction of their team leader's goals, even when the leader was not there. For example, privileged developers were seen as instrumental in establishing and enforcing effective coding standards and acceptance criteria for ISD tasks. They set good examples by frequently engaging in code refactoring to achieve high software quality in their team's product. In meetings, several team leaders relied on key individuals to act as unofficial proxies for them. As one team leader noted: "I have a few developers I can count on to fill in effectively for me and I really do use that when needed."

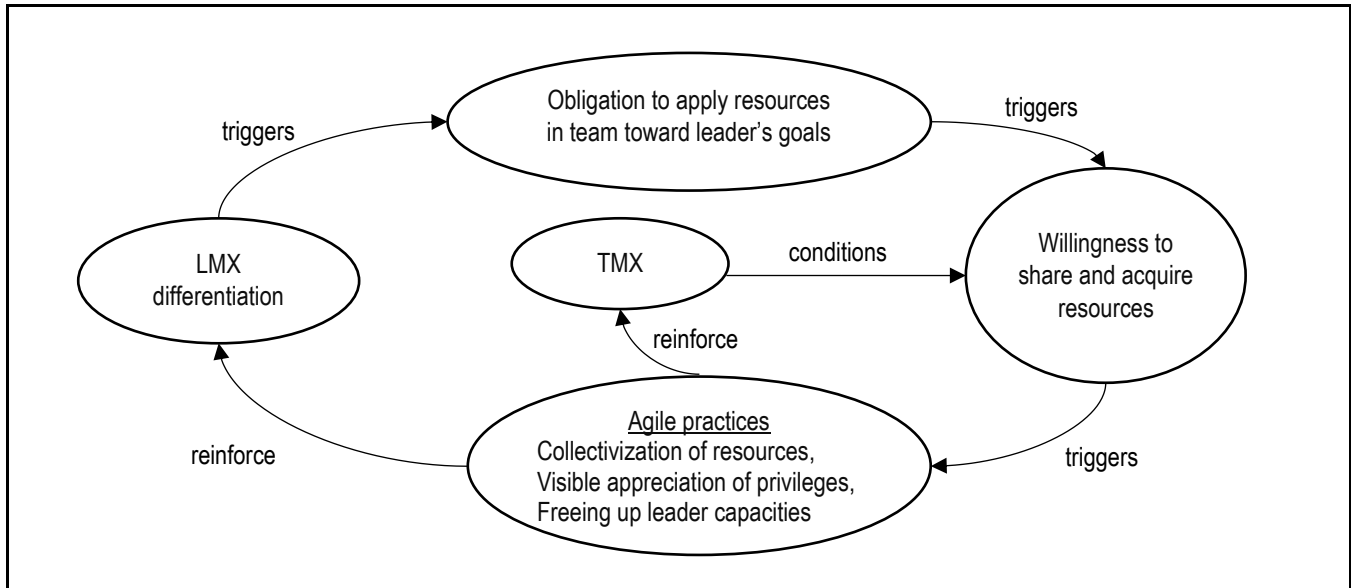
Across the teams with high TMX and LMX differentiation, all team leaders agreed that such self-organization "created much more flexibility" in their role as a team leader and helped them to "focus on the big picture" instead of micromanagement. As a result, team leaders could use the freed-up time to acquire new resources for their team and remove external obstacles that hindered their team's success. Thus, freeing up leader capacities also had beneficial effects for the teams.

### Meta-Inferences

Our survey results suggest that TMX plays a critical role in whether LMX differentiation is associated with higher or

lower developer satisfaction. Our case study results suggest that teams leverage and reinforce LMX differentiation by enacting three mechanisms through agile practices. Together, these results point toward the existence of a long-term reinforcement cycle of LMX differentiation in agile ISD teams that hinges on two factors: (1) the existing exchange relationships between developers in the form of TMX and (2) the enactment of social exchange through agile practices. Figure 3 depicts this reinforcement cycle.

When team leaders engage in high LMX differentiation, they provide some privileged developers with more valuable resources. Consequently, the privileged developers feel an obligation to reciprocate these actions. But privileged team members cannot repay team leaders with equal currency due to their different roles. Instead, they feel obliged to use the received benefits to contribute to achieving their team leader's goals. In agile ISD teams, these goals include effective self-organization and collaboration. Thus, privileged developers are willing to share resources with their nonprivileged team members in meetings and by engaging in collaborative agile practices to send a signal to the team leader (e.g., I am actively using my privileges for the benefit of the team). Likewise, nonprivileged team members are willing to engage in agile practices with their privileged colleagues, in order to acquire some of the high-quality resources to which they would otherwise have no access.



**Figure 3. Long-Term Reinforcement Cycle of LMX Differentiation in Agile ISD Teams**

The willingness of privileged members to share high-quality resources and the willingness of nonprivileged members to acquire resources through agile practices is conditioned by a team’s TMX relationships. In low-quality TMX teams, privileged developers are less willing to engage with other members in agile practices, as they presume that the effort and resources they contribute will not be reciprocated. Likewise, in low-quality TMX teams, LMX differentiation can easily lead to envy in nonprivileged team members and deter them from close collaboration with privileged colleagues. Conversely, in high-quality TMX teams, both privileged and nonprivileged team members can trust in the reciprocity of their actions. When TMX quality is high, developers are consequently more willing to engage in effortful and intensive agile practices with their team members to acquire and share resources. They are more likely to use agile practices that effectively distribute resources across the team through the collectivization of resources, visible appreciation of the team leader, and freeing up leader capacities. By engaging in these practices, high-LMX developers reciprocate received benefits to their team leader and gain benefits for themselves as well as their team, thus strengthening the feelings of reciprocity between privileged and nonprivileged team members. Ultimately, in the long run, the strengthened obligations and expectations of reciprocity reinforce both TMX relationships and LMX differentiation.

Although not intended as a test, our two focus groups provided some corroboration of the integrated results and suggested that the reinforcement cycle of LMX differentiation was present in the target population. Appendix F provides details about the focus group participants, procedures, and results.

## Discussion

In sum, our work aimed to understand the effects of LMX differentiation on agile ISD team members and the processes through which ISD teams address it. We used SET to theorize the effects of LMX differentiation on developer job satisfaction and developer team satisfaction. We found support for our hypotheses that there are positive associations of LMX differentiation with developer satisfaction that are particularly strong when teams have high-quality TMX. We elicited three mechanisms through which teams address LMX differentiation using agile practices (i.e., collectivization of resources, visible appreciation of privileges, freeing up leader capacities). Together, our findings lend support to the reasoning that ISD teams rely on context-specific team processes that help all members to reap the benefits of LMX differentiation and reinforce TMX as well as LMX differentiation.

## Theoretical Implications

This work has three key theoretical implications. First, prior research on leadership in ISD teams holds that team leaders establish interpersonal control relationships so that developers act more effectively and develop more desirable psychological states (Venkatesh et al., 2018; Windeler et al., 2017), but it has not accounted for the possibility that team leaders may differentiate such interpersonal relationships across team members. We add to this literature by showing that the way in which ISD team leaders engage in

interpersonal relationships with their individual subordinates must be viewed in light of the overall team. LMX differentiation enables team leaders to efficiently interact with the larger team by leaving routine challenges and issues of low-performing developers to the self-organizing team and providing more valuable resources to key team members. In teams with high-quality TMX relationships, these resources are shared through agile practices for the benefit of the whole team, thus resulting in developers being more satisfied with their jobs and teams. Future research may need to account for the possibility that other aspects of leadership in ISD, such as the elements of empowering leadership (Windeler et al., 2017) or internal process control (Venkatesh et al., 2018), are not equally directed toward all team members either.

Second, we provide a context-specific explanation of how team processes—in the form of three mechanisms embedded in collaborative agile practices—allow agile ISD teams to leverage the high-quality resources of differentiated LMX relationships for the benefit of the overall team. We show that collaborative agile practices allow all team members to benefit from LMX differentiation. Such team processes may be the reason why prior work has found that ISD teams sometimes benefit from LMX differentiation whereas most other professions do not (Yu et al., 2018). Our work provides a springboard for future research investigating how to feed our context-specific results back into theory on LMX differentiation, e.g., by investigating how the group processes of collectivization of resources, visible appreciation of privileges, and freeing up leader capacities can be captured, transferred to, and tested in contexts other than agile ISD.

Third, we contribute to the literature on LMX differentiation by showing that TMX is a key moderator of the cross-level effects of LMX differentiation on developers' satisfaction with their jobs and teams, at least for agile ISD teams. Prior work has provided inconclusive results regarding the cross-level effects of LMX differentiation on team members (Anand et al., 2014). Our study highlights TMX as a key factor that influences whether LMX differentiation is associated with more or less developer job and team satisfaction. High-LMX team members are more willing to share valuable resources with their team members if they can count on reciprocity due to high-quality TMX relationships. They can provide their team members with access to otherwise inaccessible resources that contribute to their team members' job and team satisfaction.

### **Limitations and Future Research**

This study has some limitations that provide avenues for future research. First, we collected data from a single Indian

software company. Future research could extend our study to additional companies and different countries. Second, the company relied mainly on XP, and its developers addressed LMX differentiation with XP practices. Future research could study teams that use other agile ISD methods to see how they respond to LMX differentiation. Finally, we used SET to understand LMX differentiation and related team processes, thereby focusing on resource exchange, obligation, and reciprocity. Prior work on LMX differentiation has also applied other theories, such as social comparison and relative deprivation theories (Yu et al., 2018, p. 1160), and future research might consider these theories in order to elaborate whether the team processes we identified possibly have more functions in dealing with LMX differentiation than those related to social exchange.

### **Practical Implications**

This work sheds light on how to manage the relationship between team leaders and developers in agile ISD teams. First, we show that equal treatment of all team members is not necessarily the best option. Instead, team leaders may wish to spend more effort managing relationships with some key team members and provide them with the highest-quality information, advice, and social support. This insight is particularly important in light of reports that agile ISD teams tend to treat all team members equally, sometimes to the point that competency differences are no longer reflected (Drury et al., 2012). Second, team leaders should consider the quality of TMX within their teams. In teams with high-quality TMX, the resources provided by leaders to key team members add much value, as they can also be leveraged by other team members during agile ISD. By conveying the expectation that key team members should share and augment their resources through collaboration and self-organization for the good of the team, team leaders can further foster beneficial team processes as well as performance outcomes. For teams with low-quality TMX, in contrast, team leaders should establish homogeneous LMX with all team members. Otherwise, the privileges granted to key members may increase inequality in the team, exacerbate the risk of team conflict, and reduce developers' willingness to engage in intensive collaboration for the benefit of the team.

### **Acknowledgments**

We thank the senior editor, associate editor, and five reviewers for their constructive comments. This research was partially funded by a Hong Kong Research Grants Council grant (693313).

## References

- 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*(3), 627-639. <https://doi.org/10.1016/j.infsof.2014.09.002>
- Akaike, H. (1974) A new look at the statistical model identification. *IEEE Transactions on Automatic Control, 19*(6), 716-723. <https://doi.org/10.1109/TAC.1974.1100705>
- Anand, S., Vidyarthi, P. R., Liden, R. C., & Rousseau, D. M. (2010). Good citizens in poor-quality relationships: Idiosyncratic deals as a substitute for relationship quality. *Academy of Management Journal, 53*(5), 970-988. <https://doi.org/10.5465/amj.2010.54533176>
- Anand, S., Vidyarthi, P., & Park, H. S. (2014). LMX differentiation: Understanding relational leadership at individual and group levels. In Bauer, T. and Erdogan, B. (eds.), *The Oxford handbook of leader-member exchange* (pp. 263-292). Oxford University Press. <https://doi.org/10.1093/oxfordhb/9780199326174.013.0007>
- Banks, G. C., Batchelor, J. H., Seers, A., O'Boyle Jr, E. H., Pollack, J. M., & Gower, K. (2014). What does team-member exchange bring to the party? A meta-analytic review of team and leader social exchange. *Journal of Organizational Behavior, 35*(2), 273-295. <https://doi.org/10.1002/job.1885>
- Bauer, D. J., Preacher, K. J., & Gil, K. M. (2006). Conceptualizing and testing random indirect effects and moderated mediation in multilevel models: New procedures and recommendations. *Psychological Methods, 11*(2), 142-163. <https://doi.org/10.1037/1082-989X.11.2.142>
- Bick, S., Spohrer, K., Hoda, R., Scheerer, A., & Heinzl, A. (2018). Coordination challenges in large-scale software development: A case study of planning misalignment in hybrid settings. *IEEE Transactions on Software Engineering, 44*(10), 932-950. <https://doi.org/10.1109/TSE.2017.2730870>
- Blau, P. M. (1964). *Exchange and power in social life*. New York: Wiley.
- Bliese, P. (2016). Multilevel modeling in R (2.6). Retrieved on January 20, 2020 from [http://rsync.udc.es/CRAN/doc/contrib/Bliese\\_Multilevel.pdf](http://rsync.udc.es/CRAN/doc/contrib/Bliese_Multilevel.pdf).
- Bliese, P. D., & Hanges, P. J. (2004). Being Both Too Liberal and Too Conservative: The Perils of Treating Grouped Data as Though They Were Independent. *Organizational Research Methods, 7*(4), 400-417. <https://doi.org/10.1177/1094428104268542>
- Chan, F. K. Y., & Thong, J. Y. L. (2009). Acceptance of agile methodologies: A critical review and conceptual framework. *Decision Support Systems, 46*(4), 803-814. <https://doi.org/10.1016/j.dss.2008.11.009>
- Chen, G., Smith, T. A., Kirkman, B. L., Zhang, P., Lemoine, G. J., & Farh, J.-L. (2019). Multiple team membership and empowerment spillover effects: Can empowerment processes cross team boundaries? *Journal of Applied Psychology, 104*(3), 321-340. <https://doi.org/10.1037/apl0000336>
- Cropanzano, R., & Mitchell, M. S. (2005). Social exchange theory: An interdisciplinary review. *Journal of Management, 31*(6), 874-900. <https://doi.org/10.1177/0149206305279602>
- Drury, M., Conboy, K., & Power, K. (2012). Obstacles to decision making in agile software development teams. *Journal of Systems and Software, 85*(6), 1239-1254. <https://doi.org/10.1016/j.jss.2012.01.058>
- Epitropaki, O., Kapoutsis, I., Ellen, B. P., Ferris, G. R., Drivas, K., & Ntotsi, A. (2016). Navigating uneven terrain: The roles of political skill and LMX differentiation in prediction of work relationship quality and work outcomes. *Journal of Organizational Behavior, 37*(7), 1078-1103. <https://doi.org/10.1002/job.2100>
- Erdogan, B., & Bauer, T. (2010). Differentiated leader-member exchanges: The buffering role of justice climate. *Journal of Applied Psychology, 95*(6), 1104-1120. <https://doi.org/10.1037/a0020578>
- Faraj, S., & Sambamurthy, V. (2006). Leadership of information systems development projects. *IEEE Transactions on Engineering Management, 53*(2), 238-249. <https://doi.org/10.1109/TEM.2006.872245>
- Farh, C. I., Lanaj, K., & Ilies, R. (2017). Resource-based contingencies of when team-member exchange helps member performance in teams. *Academy of Management Journal, 60*(3), 1117-1137. <https://doi.org/10.5465/amj.2014.0261>
- Farmer, S. M., Van Dyne, L., & Kamdar, D. (2015). The contextualized self: How team-member exchange leads to coworker identification and helping OCB. *Journal of Applied Psychology, 100*(2), 583-595. <https://doi.org/10.1037/a0037660>
- Frank, K. A. (2000). Impact of a confounding variable on a regression coefficient. *Sociological Methods and Research, 29*(2), 147-194. <https://doi.org/10.1177/0049124100029002001>
- Frank, K. A., Maroulis, S. J., Duong, M. Q., & Kelcey, B. M. (2013). What would it take to change an inference? Using Rubin's causal model to interpret the robustness of causal inferences. *Educational Evaluation and Policy Analysis, 35*(4), 437-460. <https://doi.org/10.3102/0162373713493129>
- Ghobadi, S., & Mathiassen, L. (2017). Risks to effective knowledge sharing in agile software teams: A model for assessing and mitigating risks. *Information Systems Journal, 27*(6), 699-731. <https://doi.org/10.1111/isj.12117>
- Graen, G. B., & Uhl-Bien, M. (1995). Relationship-based approach to leadership: Development of leader-member exchange (LMX) theory of leadership over 25 years: Applying a multi-level multi-domain perspective. *The Leadership Quarterly, 6*(2), 219-247. [https://doi.org/10.1016/1048-9843\(95\)90036-5](https://doi.org/10.1016/1048-9843(95)90036-5)
- Herdman, A. O., Yang, J., & Arthur, J. B. (2017). How does leader-member exchange disparity affect teamwork behavior and effectiveness in work groups? The moderating role of leader-leader exchange. *Journal of Management, 43*(5), 1498-1523. <https://doi.org/10.1177/0149206314556315>
- Hong, W., Chan, F. K. Y., Thong, J. Y. L., Chasalow, L. C., & Dhillon, G. (2014). A framework and guidelines for context-specific theorizing in information systems research. *Information Systems Research, 25*(1), 111-136. <https://doi.org/10.1287/isre.2013.0501>
- Hong, W., Thong, J. Y. L., Chasalow, L., & Dhillon, G. (2011). User acceptance of agile information systems: A model and empirical



- test. *Journal of Management Information Systems*, 28(1) 235-272. <https://doi.org/10.2753/MIS0742-1222280108>
- Hooper, D. T., & Martin, R. (2008). Beyond personal leader-member exchange (LMX) quality: The effects of perceived LMX variability on employee reactions. *The Leadership Quarterly*, 19(1), 20-30. <https://doi.org/10.1016/j.leaqua.2007.12.002>
- Hsieh, J., Rai, A., & Xu, S. X. (2011). Extracting business value from IT: A sensemaking perspective of post-adoptive use. *Management Science*, 57(11), 2018-2039. <https://doi.org/10.1287/mnsc.1110.1398>
- Huang, P., Tafti, A. R., & Mithas, S. (2018). Platform sponsor's investments and user contributions in knowledge communities: The role of knowledge seeding. *MIS Quarterly*, 42(1), 213-240. <https://doi.org/10.25300/MISQ/2018/13490>
- Igbaria, M., Zinatelli, N., Cragg, P., & Cavaye, A. L. M. (1997). Personal computing acceptance factors in small firms: A structural equation model. *MIS Quarterly*, 21(3), 279-305. <https://doi.org/10.2307/249498>
- Johnson, B., Zimmermann, T., & Bird, C. (2021). The effect of work environments on productivity and satisfaction of software engineers. *IEEE Transactions on Software Engineering*, 47(4), 736-757. <https://doi.org/10.1109/TSE.2019.2903053>
- Khanagha, S., Volberda, H. W., Alexiou, A., & Annosi, M. C. (2022). Mitigating the dark side of agile teams: peer pressure, leaders' control, and the innovative output of self-managing teams. *Journal of Product Innovation Management*, 39(3), 334-350. <https://doi.org/10.1111/jpim.12589>
- Kude, T., Mithas, S., Schmidt, C. T., & Heinzl, A. (2019). How pair programming influences team performance: The role of backup behavior, shared mental models, and task novelty. *Information Systems Research*, 30(4), 1145-1163. <https://doi.org/10.1287/isre.2019.0856>
- Li, A. N., & Liao, H. (2014). How do leader-member exchange quality and differentiation affect performance in teams? An integrated multilevel dual process model. *Journal of Applied Psychology*, 99(5), 847-866. <https://doi.org/10.1037/a0037233>
- Lindell, M. K., & Whitney, D. J. (2001). Accounting for common method variance in cross-sectional research designs. *Journal of Applied Psychology*, 86(1), 114-121. <https://doi.org/10.1037/0021-9010.86.1.114>
- Mani, D., Barua, A., & Whinston, A. B. (2012). An empirical analysis of the contractual and information structures of business process outsourcing relationships. *Information Systems Research*, 23(3), 618-634. <https://doi.org/10.1287/isre.1110.0374>
- Martin, R., Thomas, G., Legood, A., & Dello Russo, S. (2018). Leader-member exchange (LMX) differentiation and work outcomes: Conceptual clarification and critical review. *Journal of Organizational Behavior*, 39(2), 151-168. <https://doi.org/10.1002/job.2202>
- Maruping, L. M., & Magni, M. (2015). Motivating employees to explore collaboration technology in team contexts. *MIS Quarterly*, 39(1), 1-16.
- Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research*, 20(3), 377-399. <https://doi.org/10.1287/isre.1090.0238>
- Maruping, L. M., Venkatesh, V., Thong, J. Y. L., & Zhang, X. (2019). A risk mitigation framework for information technology projects: A cultural contingency perspective. *Journal of Management Information Systems*, 36(1), 120-157. <https://doi.org/10.1080/07421222.2018.1550555>
- Mathieu, J. E., & Chen, G. (2011). The etiology of the multilevel paradigm in management research. *Journal of Management* 37(2), 610-641. <https://doi.org/10.1177/0149206310364663>
- Miles, M. B., & Huberman, M. A. (1994). *Qualitative data analysis: An expanded sourcebook*. SAGE.
- Morris, M. G., & Venkatesh, V. (2010). Job characteristics and job satisfaction: Understanding the role of enterprise resource planning system implementation. *MIS Quarterly*, 34(1), 143-161. <https://doi.org/10.2307/20721418>
- PMI (2017). *Agile practice guide*. Project Management Institute.
- Podsakoff, P. M., Scott, B. M., Lee, J. Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. *Journal of Applied Psychology*, 88(5), 879-903. <https://doi.org/10.1037/0021-9010.88.5.879>
- Rai, A., Maruping, L. M., & Venkatesh, V. (2009). Offshore information systems project success: The role of social embeddedness and cultural characteristics. *MIS Quarterly*, 33(3), 617-649. <https://doi.org/10.2307/20650313>
- Ramasubbu, N., & Bardhan, I. (2021). Reconfiguring for agility: Examining the performance implications for project team autonomy through an organizational policy experiment. *MIS Quarterly*, 45(4), 2261-2280. <https://doi.org/10.25300/MISQ/2021/14997>
- Ramasubbu, N., Bharadwaj, A., & Tayi, G. K. (2015). Software process diversity: Conceptualization, measurement, and analysis of impact on project performance. *MIS Quarterly*, 39(4), 787-807.
- Saldaña, J. (2013). *The coding manual for qualitative researchers*, (2nd ed.). SAGE.
- Sarker, S., & Sarker, S. (2009). Exploring agility in distributed information systems development teams: An interpretive study in an offshoring context. *Information Systems Research*, 20(3), 440-461. <https://doi.org/10.1287/isre.1090.0241>
- Sarker, S., Xiao, X., Beaulieu, T., & Lee, A. S. (2018). Learning from first-generation qualitative approaches in the IS discipline: An evolutionary view and some implications for authors and evaluators. *Journal of the Association for Information Systems*, 19(10), 752-774. <https://doi.org/10.17705/1jais.00508>
- Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 461-464.
- Seers, A., Petty, M. M., & Cashman, J. F. (1995). Team-member exchange under team and traditional management. *Group & Organization Management*, 20(1), 18-38. <https://doi.org/10.1177/1059601195201003>
- Shastri, Y., Hoda, R., & Amor, R. (2021a). Spearheading agile: The role of the scrum master in agile projects. *Empirical Software Engineering*, 26(1), 1-31. <https://doi.org/10.1007/s10664-020-09899-4>
- Shastri, Y., Hoda, R., & Amor, R. (2021b). The role of the project

- manager in agile software development projects. *Journal of Systems and Software*, 173, Article 110871. <https://doi.org/10.1016/j.jss.2020.110871>
- Snijders, T. A. B., & Bosker, R. J. (1999). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. SAGE.
- Stewart, D. W., & Shamdasani, P. N. (2014). *Focus groups: Theory and practice*. SAGE.
- Storey, M. A., Zimmermann, T., Bird, C., Czerwonka, J., Murphy, B., & Kalliamvakou, E. (2019). Towards a theory of software developer job satisfaction and perceived productivity. *IEEE Transactions on Software Engineering*, 47(10), 2125-2142. <https://doi.org/10.1109/TSE.2019.2944354>
- Trauth, E. M., & Jessup, L. M. (2000). Understanding computer-mediated discussions: Positivist and interpretive analyses of group support system use. *MIS Quarterly*, 24(1), 43-79. <https://doi.org/10.2307/3250979>
- Tripp, J. F., Riemenschneider, C., & Thatcher, J. B. (2016). Job Satisfaction in agile development teams: agile development as work redesign. *Journal of the Association for Information Systems*, 17(4), 267-307. <https://doi.org/10.17705/1jais.00426>
- Venkatesh, V., Brown, S., & Bala, H. (2013). Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems. *MIS Quarterly*, 37(1), 21-54.
- Venkatesh, V., Brown, S., & Sullivan, Y. (2016). Guidelines for conducting mixed-methods research: An extension and illustration. *Journal of the Association for Information Systems*, 17(7), 435-494. <https://doi.org/10.17705/1jais.00433>
- Venkatesh, V., Shaw, J. D., Sykes, T. A., Wamba, S. F., & Macharia, M. (2017). Networks, technology, and entrepreneurship: A field quasi-experiment among women in rural India. *Academy of Management Journal*, 60(5), 1709-1740. <https://doi.org/10.5465/amj.2015.0849>
- Venkatesh, V., Rai, A., & Maruping, L. M. (2018). Information systems projects and individual developer outcomes: Role of project managers and process control. *Information Systems Research*, 29(1), 127-148. <https://doi.org/10.1287/isre.2017.0723>
- Venkatesh, V., Thong, J. Y. L., Chan, F. K. Y., Hoehle, H., & Spohrer, K. (2020). How agile software development methods reduce work exhaustion: Insights on role perceptions and organizational skills. *Information Systems Journal*, 30(4), 733-761. <https://doi.org/10.1111/isj.12282>
- Windeler, J., Maruping, L. M., & Venkatesh, V. (2017). Technical systems development risk factors: The role of empowering leadership in lowering developers' stress. *Information Systems Research*, 28(4), 775-796. <https://doi.org/10.1287/isre.2017.0716>
- Wunderlich, P., Veit, D., & Sarker, S. (2019). Adoption of sustainable technologies: A mixed-methods study of German households. *MIS Quarterly*, 43(2), 673-691. <https://doi.org/10.25300/MISQ/2019/12112>
- Yu, A., Matta, F. K., & Cornfield, B. (2018). Is leader-member exchange differentiation beneficial or detrimental for group effectiveness? A meta-analytic investigation and theoretical integration. *Academy of Management Journal*, 61(3), 1158-1188. <https://doi.org/10.5465/amj.2016.1212>

## About the Authors

**Viswanath Venkatesh**, who completed his Ph.D. at the University of Minnesota, is an Eminent Scholar and Verizon Chair at the Pamplin College of Business, Virginia Tech. Since Fall 2021, he is also the Director of Pamplin's Executive Ph.D. program. He is widely regarded as one of the most influential scientists, both in terms of premier journal publications and citation impact (e.g., Thomson Reuters' highlycited.com, Emerald Citations, SSRN, PLoS Biology), with a recent career ranking of 485th (out of ~9 million scientists in all fields) and 1st in information systems. His 2021 citation impact ranking was 95th (out of ~9 million scientists in all fields). His research focuses on understanding the diffusion of technologies in organizations and society. His favorite project focuses on improving the quality of life of the poorest of the poor in India—which he has presented in various forums including at the United Nations. The sponsorship of his research has been about US\$10M. His work has appeared in leading journals in human-computer interaction, information systems, organizational behavior, psychology, marketing, medical informatics, and operations management, and included best paper awards (e.g., *Academy of Management Journal*). His works have been cited about 170,000 times (Google Scholar) and over 56,000 times (Web of Science), with an h-index of 88 and i-10 index of 147. He developed and maintains an IS research rankings website that has received many accolades from the academic community including Association for Information Systems' Technology Legacy Award. He has served in editorial roles in various journals. He is a Fellow of the AIS and the Information Systems Society, INFORMS.

**James Y. L. Thong** is the Michael Jepsen Professor of Business and Chair Professor of Information Systems at the School of Business and Management, Hong Kong University of Science and Technology. He received his Ph.D. from the National University of Singapore. His research on technology adoption, e-government, human-computer interaction, information privacy and security, and IT development teams has appeared in *MIS Quarterly*, *Information Systems Research*, *Journal of Management Information Systems*, and *Journal of the Association for Information Systems*, among other outlets, with 36,000+ Google Scholar citations and 17,000+ SCOPUS citations. He has served as a senior editor and associate editor of *MIS Quarterly*, and an associate editor for *Information Systems Research*. He received the *ISR* Best Associate Editor award and was named an AIS Fellow. ORCID: <http://orcid.org/0000-0002-1640-0581>

**Kai Spohrer** is an associate professor of information systems at the Frankfurt School of Finance & Management, Germany. He received his Ph.D. in information systems from the University of Mannheim for his dissertation on collaborative quality assurance and team cognition in software development teams. He completed his Habilitation at the University of Mannheim with research on digital innovation in healthcare and the software industry. His research aims to understand and improve the development and use of information systems, especially collaboration and coordination in healthcare and the software industry. His work has appeared and

is forthcoming in leading journals in information systems and software engineering, including *MIS Quarterly*, *Information Systems Research*, *IEEE Transactions of Software Engineering*, and *Journal of Management Information Systems*. ORCID: <https://orcid.org/0000-0001-8659-7554>

**Frank K. Y. Chan** is a professor of information systems in the Department of Information Systems, Decision Sciences and Statistics at ESSEC Business School, France. He received his Ph.D. from the Hong Kong University of Science and Technology. His research interests include electronic government and technology implementation. His work has appeared in *MIS Quarterly*, *Information Systems Research*, *Journal of the Association for Information Systems*, *Information Systems Journal*, *Journal of Operations Management*, *Journal of Business Ethics*, and *Public Administration Review*. ORCID: <https://orcid.org/0000-0001-9301-7634>

**Ankur Arora** is an assistant professor of business information & technology in the Fogelman College of Business and Economics at the University of Memphis. His research primarily focuses on the development, management, and impact of organizational artificial

intelligence systems, and applications. His scholarly work has been published, or is under review, in outlets such as *Information Systems Research*, *IT & People*, and the *International Journal of Medical Informatics*. He received his Ph.D. from the University of Arkansas in 2022. ORCID: <https://orcid.org/0000-0002-9502-5547>

**Hartmut Hoehle** is a professor and chair of Enterprise Systems at the University of Mannheim, Germany. He received a Ph.D. in information systems from Victoria University of Wellington, New Zealand. His research interests include the design, implementation, and use of enterprise systems. Stemming from his professional experiences gained while working at Deutsche Bank, he is particularly interested in how services and products can be distributed through electronically mediated channels. His work has appeared in *MIS Quarterly*, *Journal of Management Information Systems*, *European Journal of Information Systems*, *Decision Support Systems*, *International Journal of Operations & Production Management*, and other journals. ORCID: <https://orcid.org/0000-0001-8117-0105>

**Srinivasan Venkatraman** is the chief data scientist at The Boeing Company.

# Appendix A

## Quality Criteria Evaluation for Our Mixed Methods Study

**Table A1. Quality Criteria of Mixed Methods Study (adapted from Venkatesh et al., 2013 and Wunderlich et al., 2019)**

Aspect	Quality criteria	Implementation in this study
Design quality	Design Suitability	<b>Survey</b> study addressed RQ1, testing the statistically relevant association of LMX differentiation with developer satisfaction and the moderating effect of TMX in agile ISD teams across contexts.
		<b>Embedded case study</b> addressed RQ2, unearthing previously unknown team processes that underlie associations of LMX differentiation with developer satisfaction from rich interview data.
		<b>Together</b> , the survey and case study aimed at providing a more complete picture of how agile ISD teams leverage LMX differentiation to benefit their team members.
	Design adequacy	<b>Survey</b> followed established guidelines and procedures. Collected data in multiple project stages from different sources at team and individual levels. Used established scales and meaningful control variables.
		<b>Case study</b> followed established guidelines and procedures. Used leader and developer interviews as distinct sources of evidence; created a sample that covered all combinations of high/low TMX and LMX differentiation. Conducted case study in an environment where we already knew much about the organizational context (same software company as survey). Collected field notes during interviews; used SET as a guiding frame but remained open for results emerging from data.
	Analytic adequacy	<b>Survey:</b> RCM analyses followed guidelines and examples from premier journals in IS and related disciplines. Conducted various robustness checks, including tests against nonresponse bias, common method bias, reverse causality, selection bias, omitted variable bias.
<b>Case study:</b> Analysis followed guidelines for versus-coding and embedded case studies, comparing responses of low-LMX developers to those of high-LMX developers (within-case analysis) and responses from teams with low/medium TMX to those of high-TMX teams (cross-case analysis). Codes developed by one author were discussed and refined with other authors. Built a chain of evidence by cross-validating qualitative assessments of TMX and LMX differentiation with survey results from case teams. Used abductive reasoning with SET as a guiding theoretical frame. Samples of coding and data structure provided. Quotes provided to reflect rich context.		
Explanation quality	Quantitative inferences	<p><b>Internal validity:</b> Ensured convergent and discriminant validity as well as reliability of constructs. Ensured aggregation validity for convergent team-level measures. Reported various descriptives.</p> <p><b>Statistical conclusion validity:</b> Applied RCM as a recognized technique for multilevel data analysis. Conducted and reported various robustness checks.</p> <p><b>External validity:</b> Checked sampling frame against sample and found no deviations. Drew a relatively large sample with more than 200 projects and more than 1900 developers. Limitations regarding single company and single national culture were acknowledged.</p>

	Qualitative inferences	<p><b>Authenticity:</b> Leveraged direct voices of developers (satisfied/dissatisfied with low/high LMX, in teams with low/high TMX). Developed understanding for developer opinions about LMX differentiation drawing on their embeddedness in a team structure.</p> <p><b>Plausibility:</b> All inferences were double-checked by all authors and found plausible by participants in focus groups. All inferences are compatible with the general ideas of SET.</p> <p><b>Reconciliation of polyphonic narrative:</b> Understanding TMX as a conditional factor and team processes as emergent from LMX differentiation and TMX allowed us to understand why some low-LMX developers were very unhappy about LMX differentiation, whereas other low-LMX developers were completely fine with not being one of the privileged team members with high LMX.</p>
	Integrative inferences	<p><b>Integrative efficacy:</b> Survey and case study results were integrated into a reinforcement cycle that is theoretically consistent and compatible with the survey results, case study results, and the general ideas of SET.</p> <p><b>Inference transferability:</b> Focus groups were applied to make sure that the findings were transferable to teams within the target population of ISD teams. The findings may be transferable to other teamwork contexts but our context of agile ISD teams in a single firm did not allow for empirical generalization. We call for future work to do so.</p> <p><b>Integrative correspondence:</b> Understanding the inferred reinforcement cycle provides a more complete picture of how ISD team members come to benefit from LMX differentiation than survey results or case study results could provide in isolation.</p>

# Appendix B

## Survey Items

All constructs (except as indicated) were measured using a 7-point Likert scale ranging from 1 (“strongly disagree”) to 7 (“strongly agree”).

### **Developer team satisfaction** (Windeler et al., 2017)

Please rate your team on the following dimensions:

1. The team works efficiently.
2. The team produces high quality technical innovations.
3. The team adheres to schedules.
4. The team adheres to budgets.
5. The team demonstrates an ability to resolve conflicts.

### **Developer job satisfaction** (Morris & Venkatesh, 2010)

1. Overall, I am satisfied with my job.
2. I would prefer another, more ideal job. (reverse coded)
3. I am satisfied with the important aspects of my job.

### **Leader-member exchange (LMX)** (Graen & Uhl-Bien, 1995)

1. How well does your leader understand your job problems and needs? <not a bit; a little; a fair amount; quite a bit; a great deal>
2. How well does your leader recognize your potential? <not at all; a little; moderately; mostly; fully>
3. Regardless of how much formal authority he/she has built into his/her position, what are the chances that your leader would use his/her power to help you solve problems in your work? <none; small; moderate; high; very high>
4. Do you know where you stand with your leader and do you usually know how satisfied your leader is with what you do? <rarely; occasionally; sometimes; fairly often; very often>
5. Again, regardless of the amount of formal authority your leader has, what are the chances that he/she would “bail you out,” at his/her expense? <none; small; moderate; high; very high>
6. I have enough confidence in my leader that I would defend and justify his/her decision if he/she were not present to do so? <strongly disagree; disagree; neutral; agree; strongly agree>
7. How would you characterize your working relationship with your leader? <extremely ineffective; worse than average; average; better than average; effective>

### **Team-member exchange (TMX)** (Seers et al., 1995) (Likert scale from 1 (“never”) to 7 (“always”))

1. How willing are other members of your team to help finish work that was assigned to you?
2. How willing are you to help finish work that had been assigned to others?
3. How flexible are you about switching job responsibilities to make things easier for other team members?
4. How well do other members of your team recognize your potential?
5. How well do other members of your team understand your problems and needs?
6. How often do you make suggestions about better work methods to other team members?
7. In busy situations, how often do other team members ask you to help out?
8. Do other members of your team usually let you know when you do something that makes their jobs easier (or harder)?
9. How often do you let other team members know when they have done something that makes your job easier (or harder)?
10. In busy situations, how often do you volunteer your efforts to help others on your team?

### **Requirements volatility** (Maruping et al., 2009)

1. Requirements fluctuated quite a bit in early phases of this project.
2. Requirements fluctuated quite a bit in later phases of this project.
3. Requirements identified at the beginning of the project were quite different from those toward the end.

### **Project complexity** (Rai et al., 2009)

The number of adjusted function points, which adjusts the count of function points by the total ratings of 14 complexity characteristics that account for the different kinds of system requirements and development environments.

### **Agile ISD training** (adapted from Igbaria et al., 1997)

1. My organization provides adequate training for using Extreme Programming (XP).
2. In my organization, training seminars are used to expose personnel to Extreme Programming (XP).
3. Training is provided when Extreme Programming (XP) is introduced in my organization.

# Appendix C

## Survey Details and Analytical Choices

Table C1 depicts the distributional details of the focal variables in our sample. The nesting of developers in teams leads to non-independence of observations and calls for testing cross-level effects. RCM calculates relationships between dependent variables at a nested lower (i.e., individual) level and independent variables at a higher (i.e., team) level (Bauer et al., 2006; Bliese & Hanges, 2004). RCM tools remedy many of the threats of type I and type II errors due to nested data by explicitly modeling non-independence (Bliese & Hanges, 2004). RCM has been used widely in IS, often for analyzing multilevel effects across projects, teams, and individuals (e.g., Maruping & Magni, 2015; Venkatesh et al., 2018; Windeler et al., 2017).

	Mean	SD	1st quartile	median	3rd quartile
LMX	4.37	1.80	5.12	4.30	3.14
TMX	4.48	1.73	5.15	4.33	3.38
LMX differentiation	3.12	1.66	4.89	3.55	2.43
Developer team satisfaction	5.51	1.38	6.12	5.29	4.30
Developer job satisfaction	5.13	1.81	5.91	5.01	4.14

**Note:**  $N = 1,894$  developers (217 teams); LMX, Developer team satisfaction, and Developer job satisfaction on level 1 (individual developer), TMX and LMX differentiation on level 2 (ISD team).

To report the explained variance that is proportionally reduced for Level 1 and Level 2 errors, we use Snijders and Bosker's (1999) overall pseudo- $R^2$  ( $\sim R^2$ ). Following prior work in IS (e.g., Windeler et al., 2017), we report the Akaike information criterion (AIC) and Bayesian information criterion (BIC) as deviance statistics that lend insight into model fit. Both measures demonstrate the trade-off between model accuracy and complexity (Akaike, 1974; Schwarz, 1978). They penalize models with superfluous parameters by adding to their measure of deviance. The BIC takes sample size into account and is a more conservative measure. Lower values for AIC and BIC indicate better model fit.

The full two-level models predicting developer job satisfaction (Model 5) and developer team satisfaction (Model 10) in Table 2 in the main body of the paper were specified according to Bliese (2016) as follows:

### Model 5, Table 2, predicting developer team satisfaction:

$$TeamSat_{ij} = \beta_{0j} + \beta_{1j}Gender + \beta_{2j}Development\ experience + \beta_{3j}LMX + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01}Team\ size + \gamma_{02}Project\ complexity + \gamma_{03}Requirements\ volatility + \gamma_{04}Agile\ ISD\ training + \gamma_{05}TMX + \gamma_{06}LMX\ differentiation + \gamma_{07}TMX \times LMX\ differentiation + U_{0j}$$

$$\beta_{1j} = \gamma_{10}; \beta_{2j} = \gamma_{20}; \beta_{3j} = \gamma_{30}$$

### Model 10, Table 2, predicting developer job satisfaction:

$$JobSat_{ij} = \beta_{0j} + \beta_{1j}Gender + \beta_{2j}Development\ experience + \beta_{3j}LMX + r_{ij}$$

$$\beta_{0j} = \gamma_{00} + \gamma_{01}Team\ size + \gamma_{02}Project\ complexity + \gamma_{03}Requirements\ volatility + \gamma_{04}Agile\ ISD\ training + \gamma_{05}TMX + \gamma_{06}LMX\ differentiation + \gamma_{07}TMX \times LMX\ differentiation + U_{0j}$$

$$\beta_{1j} = \gamma_{10}; \beta_{2j} = \gamma_{20}; \beta_{3j} = \gamma_{30}$$

Table 2 in the main body of the paper displays the results of our model testing. Of the individual-level (L1) control variables, gender and development experience were significantly associated with developer team satisfaction, whereas development experience and LMX were significantly associated with developer job satisfaction (Table 2, Models 5 and 10). When the main variables were included in the full models, the association of LMX with developer team satisfaction diminished but its association with developer job satisfaction remained significant. A potential explanation is that LMX is relatively more salient in predicting a developer's satisfaction with their own job than with their team because LMX relates to their perception of the dyadic relationship with their leader and does not concern the entire team.

Of the team-level (L2) control variables, project complexity, requirements volatility, and agile ISD training were significantly associated with developer team satisfaction, whereas team size was significantly associated with developer job satisfaction (Table 2, Models 5 and 10). On the one hand, the persistent associations of project complexity, requirements volatility, and agile ISD training with developer team satisfaction (Table 2, Models 1 to 5) confirmed previous findings on the relevance of project characteristics and training in predicting developers' perceptions about agile teams' productivity. Also, the salient effects of these project- and skill-specific variables could have accounted for the non-significance of team size, which was a general team variable and had the smallest correlation with developer team satisfaction among all control variables. On the other hand, the persistent association of team size with developer job satisfaction (Table 2, Models 6 to 10) confirmed previous findings that developers in smaller teams are more satisfied with their jobs due to less competition and the ease of creating bonds between team members (Acuña et al., 2009). It should also be noted that project complexity and requirements volatility were only significantly associated with developer job satisfaction in the control variables model (Table 2, Model 6). The results suggested that developer job satisfaction may be less influenced by project characteristics and training than developer team satisfaction is, especially when teams' social exchange relationships (LMX differentiation and TMX) are considered. A potential explanation is that with effective LMX differentiation and TMX, developers are more likely to work on tasks that match their skills and receive help from other team members, thus reducing the impacts of project characteristics and training on their job satisfaction.



# Appendix D

## Survey Robustness Checks

SET generally holds that social exchange relationships influence individuals' expectations of benefits and perceived obligations that eventually translate into behaviors such as organizational citizenship behavior and ultimately result in intra- and extra-role performance (e.g., Farmer et al., 2015). We follow the same line of argument when we predict that social exchange relationships, such as TMX and LMX differentiation, motivate developers through expected benefits and perceived obligations to actions and ultimately result in satisfaction outcomes. Nonetheless, social exchange relationships may be reinforced in the long run. We took procedural precautions and conducted several additional analyses to make our results more robust and reduce potential concerns of endogeneity.

First, we took procedural precautions in the survey design against potential reverse causality. As such, potential reverse causality may constitute a problem in our model if social exchange relationships (e.g., LMX differentiation) are caused by developer satisfaction. From a methodological perspective, the concern of potential reverse causality is much reduced by our staged survey design. As such, our independent and dependent variables were collected at different points in time at the start and end of the observed projects. This reduces concerns of reverse causality. For example, it is unlikely that developer satisfaction at the end of the project could have influenced the way leaders differentiated their LMX at the beginning of the project.

Second, consistent with prior work (e.g., Mani et al., 2012; Venkatesh et al., 2017), we conducted a two-stage Heckman procedure, as described in Hsieh et al. (2011, Appendix EC-1, "Electronic Companion"), using leader's age as an instrumental variable to correct for potential selection bias. We coded each significant, potentially endogenous variable (i.e., TMX and LMX differentiation) dichotomously as one (zero) for values above (below) the sample median. We estimated probit models to explain each of the dichotomized variables based on the control variables and developer satisfaction. We computed the inverse Mills ratio (IMR) for each of the models and added it to the original models explaining developer team and job satisfaction in Table 2 of the main body of the paper, respectively. For all variables, the results remained qualitatively unchanged after controlling for the IMR (Table D1). This reduces concerns of potential selection bias for the effects of TMX and LMX differentiation on developer team and job satisfaction.

**Table D1. Two-Stage Heckman Analysis with Instrumental Variable**

Models	Developer team satisfaction		Developer job satisfaction	
	Table 2 Model 5	Heckman 2nd Step	Table 2: Model 10	Heckman 2nd Step
<i>L1 control variables</i>				
Gender	.12* (.054)	.10* (.041)	.06 (.031)	.04 (.022)
Development experience	.13* (.051)	.10* (.044)	.11* (.048)	.10* (.045)
LMX	.10 (.14)	.08 (.044)	.11* (.053)	.10* (.044)
<i>L2 control variables</i>				
Team size	-.05 (.051)	-.04 (.033)	-.11* (.052)	-.09 (.082)
Project complexity	-.11* (.048)	-.10 (.071)	-.08 (.051)	-.06 (.077)
Requirements volatility	-.13* (.058)	-.12* (.055)	-.08 (.044)	-.05 (.120)
Agile ISD training	.12* (.053)	.11* (.047)	.10 (.080)	.08 (.104)
<i>Exchange relationships</i>				
TMX	.25*** (.029)	.21*** (.014)	.24*** (.028)	.20** (.062)
LMX differentiation	.15* (.058)	.14* (.067)	.07 (.053)	.03 (.101)
TMX x LMX differentiation	.14* (.059)	.12* (.058)	.19*** (.017)	.17** (.061)
Inverse Mills ratio		.19*** (.008)		.20** (.064)
AIC	818.37	800.28	810.41	779.43
BIC	866.84	851.62	913.69	901.02
R <sup>2</sup>	.28	.31	.27	.30

**Note:**  $N = 1,894$  developers (217 teams); leader age was used as an instrumental variable for LMX differentiation in the Heckman analysis; \*  $p < 0.05$ , \*\*  $p < 0.01$ , \*\*\*  $p < 0.001$ . Robust standard errors in parentheses.

Third, consistent with recent research on agile practices (Kude et al., 2019) and broader IS research (e.g., Huang et al., 2018), we followed the approach suggested by Frank (2000) and Frank et al. (2013) to assess the danger of possible unobserved confounding effects. We calculated the impact threshold for a confounding variable (ITCV) at which the confounding variable would render the effect of a focal independent variable (e.g., TMX) on a dependent variable (e.g., developer job satisfaction) to be nonsignificant. The ITCV determines the minimum correlations of a potential omitted variable with both the focal independent variable and the dependent variable that are necessary to render the effect of the focal independent variable on the dependent variable insignificant after controlling for all covariates. We calculated ITCV scores for the hypothesized main effects based on the main effects models predicting developer team and job satisfaction (i.e., Models 4 and 9 in Table 2, respectively). We found that the multilevel regressions of developer job satisfaction were the most sensitive to potential omitted variable bias. To invalidate our inferences on the positive associations of LMX differentiation and TMX with developer job satisfaction, omitted variables would have to have partial correlations of more than 0.66 with TMX and developer job satisfaction and partial correlations of more than 0.20 with LMX differentiation and developer job satisfaction after controlling for covariates. Given that such partial correlations are relatively high compared to the correlations in our sample, we deem it unlikely that an omitted variable can fulfill these conditions. As all other effects of focal variables were even more robust, we are confident that omitted variable bias did not constitute a threat to our results.

# Appendix E

## Case Study Data and Coding Samples

Our qualitative interviews were conducted with 35 team members from five teams and their 5 team leaders. Table E1 displays the backgrounds of the five teams as well as their levels of LMX differentiation and TMX. All five teams were engaged in developing different types of custom software such as an inventory management system (Team A) and a tool for personnel management, staffing, and development (Team E).

We first examined the interview data for hints toward our predefined codes of TMX, LMX, LMX differentiation, satisfaction, and use of agile practices. Table E2 displays a part of our coding structure and provides exemplary codes that allowed us to assess the teams' LMX differentiation and TMX as well as the team members' satisfaction. The teams varied noticeably in LMX differentiation and TMX. We went on to compare statements of team members with high LMX to those of team members with low LMX and contrasted their contexts regarding LMX differentiation and TMX. We open-coded the statements about interactions of developers that were related to their leader or to managing resources the leader provided or failed to provide. Individuals in teams with high TMX reported on a more intensive use of agile practices for activities such as knowledge sharing, indirect resource distribution, and self-organization with less involvement of the team leader than teams with low TMX. Team members with high LMX seemed to contribute differently to these activities than team members with low LMX. Table E2 displays open codes which we used to describe emerging patterns. In an abductive process that kept the core mechanisms of SET—resource exchange, obligation, and reciprocity—in mind, we aimed to make sense of these patterns. After intensive discussion in the author team using case memos and many iterations of visual mapping (Miles & Huberman, 1994), we arrived at the three team processes of collectivization of resources, visible appreciation of privileges, and freeing up leader capacities as displayed in Table E2.

Team	# Interviews members, leader	Product	LMX differentiation	TMX
A	8+1	Integrated inventory management with strategic suppliers	High	High
B	4+1	Customer relationship management with web front-end and customer portal	Low	High
C	6+1	Data warehousing and analytics engines and front-end	High	Medium
D	10+1	Web-based ERP migration	High	Low
E	7+1	Integrated HR tool for personnel management, staffing, and development	Low	Low

Predefined codes		
#	Exemplary quotes	Codes
1	Leader talks about <i>"help high-performance team members excel"</i>	High LMX differentiation
2	Leader: <i>"treating everyone equally is neither possible nor desirable."</i>	
3	Leader: <i>"I mean, you cannot treat everybody the same. You need to provide everybody with what is best for them individually and for the team as a whole. Some are fresh from university, and you can basically send them to the standard trainings [...] But then my most important team member, [...] if he comes to me to ask for resources, that can mean I have to give something exceptional that is not provided to everybody and I have and will do that."</i>	

4	<p>Developers:</p> <p><i>"Being tight with the team leads always helps you get ahead in small ways. You can get cool assignments, you can get better training opps and go to cool client locations."</i></p> <p><i>"If you are in with the team leader, you will get trained on the latest stuff including international conferences like [conference name] or other one-week Vegas training programs. If not, your *** is on the bench."</i> [edited for readability]</p> <p><i>"If my team leader likes someone, they would represent us at major meetings and even be deputed for skills needed on another team and that can help them and even us in the long-term. I mean it won't help us get promotions but it will help us be visible and help that person get a promotion."</i> [edited for readability]</p>	
5	<p>Developers:</p> <p><i>"I don't think that anybody on our team gets special treatment."</i></p> <p><i>"Our team is practically socialist."</i></p> <p><i>"We are all treated all equal by our lead."</i></p>	Low LMX differentiation
6	<p>Team member speaking of "leader's buddies" who received "prestigious tasks" to work on, as opposed to one's own "menial debugging and hot-fixing" tasks.</p>	Low LMX
7	<p>Developer: <i>"[TeamLeader] seems to see working with us as 'a necessary evil' "</i></p>	
8	<p>Developers:</p> <p><i>"I think I've got a very good relationship with [TeamLeader]. Actually, I think we all do. [TeamLeader] is very open and tries to support us however he can."</i> [edited for readability]</p> <p><i>"We all get along well with our team leader. She doesn't show any favoritism."</i></p>	High LMX, low LMX differentiation
9	<p>Developers:</p> <p><i>"Actually, I've recently been thinking about applying for positions in other teams. We used to be pretty close [in this team], but last year three colleagues left and two new ones joined the team and it is not as it used to be. [...] There is friction between some team members [...] And we also lost a lot of expertise with the leaving team members. That has not been replaced, yet. So we are struggling to do our job well."</i> [edited for readability]</p> <p><i>"I am an IC [individual contributor]. Everyone behaves like an IC. I hate it. I can't wait for this project to end."</i></p>	Low TMX, low developer team satisfaction
10	<p>Developers:</p> <p><i>"We don't really have each other's backs. Usually everyone is working for himself or herself."</i> [edited for readability]</p> <p><i>"This isn't a good group really. We come together only when we need to. We've never even been out for drinks once."</i></p>	Low TMX
11	<p>Developers:</p> <p><i>"I don't think I have ever worked on a better team."</i></p> <p><i>"Super cool team experience."</i></p>	High developer team satisfaction

Emergent codes			
#	Exemplary quotes	Open codes	Final codes
12	<p>Developers:</p> <p><i>"That's kind of a rule we follow: When you return from a training that could be of interest to anyone else it is your responsibility to share the knowledge with the rest of the team. Sometimes, we have a knowledge transfer session but often we do it throughout the day [...] And we like pair programming, so that is always a good thing."</i> [edited for readability]</p> <p><i>"We have a deal on the team that anyone who gets a cool learning opp, esp on new techs, will host a lunch at their home on a weekend and share with the rest of the team. This actually helps cut down on jealousy and backstabbing."</i></p>	<p>Knowledge sharing after training, meeting, pair programming</p>	<p><b>Collectivization of resources:</b> Privileged developers share the resources they receive from the leader with their colleagues during team task completion. Agile practices help the team members to access, share, and apply resources.</p>
13	<p>Developers:</p> <p><i>"I mean, what's the use of having only one developer who knows how to handle [new user interface technology]. All of us need to know that because all of us have to work on the UI at some point. [...] we want to retain that collective code ownership."</i> [edited for readability]</p> <p><i>"We talk and share—otherwise, we might as well abandon collective code ownership. I think we do a decent job of it now."</i></p>	<p>Knowledge sharing, collective code ownership</p>	
14	<p>Developer:</p> <p><i>"And then there are my team members [Member A] and [Member B]. They are much closer with our manager [than I am]; they can basically ask her any favor and she will try to make it happen. So, when I really have trouble and need something from our manager, I can always ask [Member A] or [Member B] to get it for me."</i></p>	<p>Indirect access through favors</p>	
15	<p>Developers:</p> <p><i>"Some of our customer meetings go better when some team members who are close to the lead start talking about one-on-ones they had with him that shaped some critical decisions. Customers often like that because it shows good group work on our side."</i></p> <p><i>"Customer meetings go best when we have our insiders driving the discussions and name-dropping often... of course, they need substance too"</i></p>	<p>Supporting leader's opinion, customer meetings, convincing team members for leader, planning meetings</p>	<p><b>Visible appreciation of privileges:</b> By visibly applying received resources, privileged developers show appreciation to the team leader and reinforce their high-quality LMX relationship.</p>
16	<p>Developers:</p> <p><i>"It [daily standups] works best when team members on the inside also sign up for boring testing assignments."</i></p> <p><i>"I don't mind not being on the inside if those on the inside also do some s*** work and help us with difficult issues."</i> [edited for readability]</p>	<p>Volunteering for chores, daily stand-up</p>	
17	<p>Leader:</p> <p><i>"That's another reason why it is so important to have at least one team member you can trust. In my team, my senior architect is usually the one who comes to me and alerts me when he realizes that the team or some team members are stuck with a problem."</i></p>	<p>Alerting the leader if necessary</p>	

18	<p>Developers:</p> <p><i>"I count on the lead's buddies to set a good example for us day in and day out."</i></p> <p><i>"It's fine if you are out drinking with the team leader all the time but better bring back cutting-edge stuff to us, especially the latest in our company coding standards, which is a moving target."</i></p> <p><i>"No one follow[s] standards or complies by the book so we need those who are the leader's favorites to bring back those things to the rest of us."</i></p>	<p>Setting good examples, coding standards, work toward team goals</p>	
19	<p>Leader:</p> <p><i>"Even if I don't have the time to attend a planning meeting or a topic discussion, I can count on them [two key developers] to set everything up as it should be."</i></p>	<p>Fulfill leader's tasks in leader's absence</p>	<p><b>Freeing up leader capacities:</b> Privileged developers create further obligations for the team leader that reinforce their high-LMX relationship. Their engagement in self-organization of the team frees up leader capacities that can be used to acquire higher quality resources for the team</p>
20	<p>Several leaders stated that the current way of working had <i>"created much more flexibility"</i> in their role as a team leader and helped them to <i>"focus on the big picture"</i></p>	<p>Leaders shift attention to other tasks</p>	
21	<p>High-LMX developer:</p> <p><i>"Fortunately, we as a team can now do many of the things that used to be done by project managers before we had XP. That is much more efficient because we can make those decisions where we have more expertise than a manager or lead or anyone higher up has."</i> [edited for readability]</p>	<p>Self-organization of former leader tasks</p>	

# Appendix F

## Focus Groups

The focus groups consisted of 10 developers and 10 team leaders from the same company who were randomly chosen from the teams and team leaders in our study. This nested sample (Venkatesh et al., 2016) helped us to ensure that the participants were all familiar with the actual developments in the project teams we studied and were, thus, able to speak to the credibility of our analyses and results. The team leaders in our leader focus group had an average tenure at the company of 8.43 years ( $SD = 1.80$ ) years, and the average tenure of the developers in our team member focus group was 3.15 years ( $SD = 1.75$ ). The average age of team leaders in the focus group was 40.11 years ( $SD = 4.33$ ), and the average age of team members in the focus group was 31.2 years ( $SD = 4.91$ ). The teams in which the participants of the team member focus group worked had an average size of 8.8 ( $SD = 1.8$ ), the team leaders in the focus group had teams of an average size of 8.8 ( $SD = 1.9$ ). The average work experience in software development in the team member focus group was 3.2 years ( $SD = 1.8$ ). Thus, the randomly drawn focus group participants fit well with the characteristics of our main sample, making them adequate judges of the credibility of the results from an internal perspective (Trauth & Jessup, 2000).

During the focus group sessions, following Stewart and Shamdasani (2014), a moderator first asked a standard set of questions to elicit participants' general reactions to the core elements of our study (i.e., LMX differentiation and how teams coped with it). We then engaged in member-checking, which is an effective triangulation strategy that examines participants' inside perspectives on the credibility of findings by reviewing analyses and interpretations with them (Trauth & Jessup, 2000). To do so, the participants were presented with the results of our survey, case study, and meta-inferences (in this order), collecting the participants' critical perspectives on each element. Participants were allowed to freely express their reactions, discuss among themselves, and provide examples. All comments were recorded and analyzed. The feedback was positive, and participants generally affirmed that the results of each step were reasonable from their perspectives. Finally, the focus group participants were presented with the elements of our meta-inferences based on exemplary statements from the studies and asked to express to which degree they found that such situations were common and important in their work life. The groups discussed each point before making their final judgment—sometimes in complete agreement and sometimes in partial agreement. The focus group sessions provided some corroboration of our results and suggested that the reinforcement cycle of LMX differentiation was present in the target population.

**Table F1. Exemplary Focus Group Results**

Concept	Importance as perceived by focus groups
LMX Differentiation	<p>Team leaders in our focus group estimated that at least 50% of other team leaders in the company deemed it more important to “help high-performance team members excel” than to treat all team members equally.</p> <p>40% of the developers in our focus group stated that they had experienced unsatisfactory situations with team leaders in which other team members with better connections to the leader had an advantage over them.</p>
Obligation to apply resources in team toward leader's goals	All 10 team leaders in the leader focus group agreed that they expected developers who received particularly valuable benefits to act in line with the leader's goals. All team leaders agreed that they would not provide any favors or special benefits to developers who did not act in line with the leader's goals.
Willingness to share & acquire resources	<p>All 10 participants in the team member focus group agreed that it was important to share resources within a team, especially if few team members had particularly valuable resources such as expert knowledge.</p> <p>All developers in the focus groups agreed that in this company it would be more beneficial for them to share high-value resources with team members than holding the resources back.</p>
Collectivization of resources	80% of the developers in our focus group agreed that the collectivization of resources happened frequently. 80% of the team leaders in the focus group agreed that the collectivization of resources was a common practice in teams of the company. All developers in the focus group agreed that the collectivization of resources happened through agile practices like pair programming and collective code ownership.

	<p>We also shared the following developer quote with the focus groups and 8 out of 10 members of each focus group endorsed the statement as something that existed in many teams of the company:</p> <p><i>“And then there are my team members [Member A] and [Member B]. They are much closer with our manager [than I am]; they can basically ask her any favor and she will try to make it happen. So, when I really have trouble and need something from our manager, I can always ask [Member A] or [Member B] to get it for me.”</i></p>
<p>Visible appreciation of privileges</p>	<p>All team leaders in our focus group agreed that visible appreciation was very common and desirable.</p> <p>80% of the developers in our focus group agreed that visible appreciation of privileges happened frequently.</p> <p>7 out of 10 developers stated that, in the past, they had made sure to show they visibly appreciated benefits received from their team leader by getting more involved in their team’s self-organizing.</p>
<p>Freeing up leader’s capacity</p>	<p>8 out of 10 team leaders agreed that they would provide more favors and support to team members who they could successfully delegate team management tasks to. All team leaders in the focus group agreed that agile development practices helped to free up some time in their schedule and allowed them to focus on the big picture of their development projects.</p> <p>9 out of 10 developers endorsed the statement that agile development practices helped teams in the company to prevent team leaders from micro-managing their teams.</p>
<p>TMX</p>	<p>8 out of 10 developers agreed that the collectivization of resources was not only beneficial for task work but also facilitated mutual trust in the reciprocation of favors between team members.</p> <p>7 out of 10 developers stated that agile development practices helped them to create a good relationship in their team in which each team member could rely on the rest of the team.</p> <p>8 out of 10 developers endorsed the statement that resource sharing in the teams of the company depended more on the relationships between individual team members than on their relationships with the team leader or management.</p>



Copyright of MIS Quarterly is the property of MIS Quarterly and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.