

Winter 12-13-2018

The Role of Leadership in Agile Software Development

Peng Xu

University of Massachusetts Boston, peng.xu@umb.edu

Yide Shen

Rowan University, Glassboro, NJ, United States., shen@rowan.edu

Follow this and additional works at: <https://aisel.aisnet.org/irwitpm2018>

Recommended Citation

Xu, Peng and Shen, Yide, "The Role of Leadership in Agile Software Development" (2018). *International Research Workshop on IT Project Management 2018*. 12.
<https://aisel.aisnet.org/irwitpm2018/12>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2018 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

The Role of Leadership in Agile Software Development

Peng Xu

College of Management
University of Massachusetts Boston
Peng.xu@umb.edu

Yide Shen

Rohrer College of Business
Rowan University
shen@rowan.edu

ABSTRACT

Agility is crucial in modern software development. Leadership is crucial for team effectiveness. Little research has been conducted to understand the role of leadership in agility in software development. In this study, we investigate the role leadership plays in achieving better performance in software development. Adopting a case study approach, we collected qualitative data to build a theoretical model that describes leadership roles in achieving team agility and improving performance. In our preliminary findings, we identified five leader roles and their leader activities. These leader functions affect two learning processes – experiential learning and vicarious learning which in turn help leaders adjust their behaviors. Such learning processes help achieve better performance by leading to a more mature, agile mentality and team capability to collectively collaborate and respond to changes in a timely manner, improving team performance.

Keywords

Agile leadership, Leadership roles, leadership activity, and Agile Software Development

INTRODUCTION

Agility is crucial in modern software development (Conboy, 2009; Conboy, Coyle, Wang, & Pikkarainen, 2011). Today, many products and their features are characterized as unique, ambiguous, and complex, as opposed to predictable, verifiable, and controllable due to dynamic changes in business environments (Moe, Dingsøyr, & Dybå, 2010). Therefore, modern software development needs flexibility and responsiveness, i.e., agility, to be successful. Agility in software development refers to a team's ability to efficiently and effectively respond to changes during the project lifecycle (Lee & Xia, 2010). Agile software development methodologies have been proposed to replace "heavyweight" methodologies in order to improve agility in software development (Magni & Maruping, 2013; Maruping, Venkatesh, & Agarwal, 2009; Sarker & Sarker, 2009).

Agile methodologies encourage coordination by self-managing teams (Gren, Torkar, & Feldt, 2017). However, the performance of such teams is challenged by a number of obstacles such as members' unwillingness to commit to or implement a decision, conflicting priorities, unstable staff, unwillingness to take ownership of decisions, conflicts between expert opinions and collaborative group opinions, conflicts between individual autonomy and team autonomy, and misguidance by leaders (Moe et al., 2010; Tessem, 2014). Researchers have cautioned managers that self-managing teams can "be difficult to implement and can risk failure when used in inappropriate situations or without sufficient leadership and support" (Moe et al., 2010, p.481).

Leadership can influence a project's objectives, motivate desired behaviors in pursuit of these objectives, change business processes, and affect group culture (Carson & Tesluk, 2007). While leadership is crucial for team effectiveness in general (Lorinkova, Pearsall, & Sims, 2013), it plays an especially critical role in maximizing the benefits of an agile methodology (Maruping et al., 2009). Without appropriate leadership and support, a team is less likely to achieve agility goals, even if it implements agile practices on paper (Moe et al., 2010). Leadership roles in agile development are different from those in traditional, plan-driven projects (Bonner, 2010). They are participants, just like other team members, with little formal authority. They are not expected to be commanders, but expected to influence the team to get things done. The strategies that used to work in traditional command-and-control environments do not work anymore. Thus, leaders in agile teams need to develop a new set of skills.

Few have studied the specific role of leadership in the context of software development agility. For example, high level job descriptions of Scrum Master and Product Owner have been outlined mainly from the practitioner literature¹. However, such high level descriptions do not provide details in application. Also, implementing agile leadership roles on paper does

¹ <https://www.scrum.org/>

not guarantee agility (Conboy et al., 2011; Moe et al., 2010). Many systems development efforts have attempted to use hybrid methods or tailored practices instead of the methods documented on paper (Vinekar, Slinkman, & Nerur, 2006). In this case, leadership roles also need to adjust their activities to address complex, dynamic challenges. Research has discussed the preferred personality type of a leader in an agile environment (Bonner, 2010). Moe and colleagues discussed the role played by a Scrum Master, but the focus was more on the team model, not the leadership roles (Moe et al., 2010). Little has been done to understand what a leader can do to build an effective team and how a leader can guide the agile process given various challenges. There is a lack of in-depth understanding on the overall leadership roles and behaviors needed to achieve software development agility.

The goal of our study is to identify effective leadership functions, which include the leadership roles and activities of each role, which can help build a software development team and enhance its agility. Using a case study methodology, we have conducted interviews with members of agile software development team at two organizations, and we continue to contact other organizations. In our preliminary findings, we categorized leadership into five leadership roles: Observer, Navigator, Initiator, External Facilitator, and Cultivator. Each role has several leadership activities that can influence a team's agility mentality and the collaborative nature of its response capability, as well as external support and coordination, which in turn supports agility in software development.

In the following sections, we discuss the theoretical background, describe our research methodology and report preliminary findings from two companies. We conclude with contributions to theory and practice.

THEORETICAL BACKGROUND

Agility in Software Development

Research has tried to define agility in software development. Some research focuses on methodology agility where agility is defined as “the continual readiness of an Information System Development [ISD] method to rapidly or inherently create change, proactively or reactively embrace change, and learn from change while contributing to perceived customer value (economy, quality, and simplicity), through its collective components and relationships with its environment ” (Conboy, 2009, p340). Others emphasize team process outcomes and define software development agility as “the software team's capability to efficiently and effectively respond to and incorporate user requirement changes during the project life cycle” (Lee & Xia, 2010, p. 90).

To achieve software development agility, various agile methodologies such as Extreme Programming (XP) and Scrum have been proposed (Lee & Xia, 2010; Moe et al., 2010). These agile methodologies propose detailed practices that a software team should follow, such as short iterations, daily meetings, frequent releases, minimal planning, and working products among others. The reported benefits of agile methods include increased productivity, faster turnaround, and higher developer satisfaction (Lindstrom & Jeffries, 2004). Though various agile methodologies differ in practices, tools, and other features, they share common principles such as an iterative approach, the embrace of changing requirements, frequent delivery, and frequent communications.

Research has attempted to explain these industry-driven agile practices via a theoretical lens (Meso & Jain, 2006; Yu & Petter, 2014). Prior research has also shown that some teams implement agile methods “on paper,” but fail to achieve agility (Conboy et al., 2011; Moe et al., 2010). In addition, increasing evidence indicates that many systems development efforts have attempted to use hybrid methods or tailored practices instead of just one method to increase agility in software development (Vinekar et al., 2006). Though agile methods propose specific agile practices, due to the complexity of software development, in-house agile practices used in each setting vary. They may bear little resemblance to any standard agile method such as XP or Scrum. These in-house agile practices fulfill the same goals as other standard agile methods using very different means (Conboy, 2009). Assessing agility in software development by strictly comparing in-house agile practices with standard practices may be difficult (Conboy, 2009). Many have focused on how to adapt standard agile practices to various environment contingencies, such as large teams and distributed software development (Bass, 2014; Goh, Pan, & Zuo, 2013; Gren, Torkar, & Feldt, 2017; Sarker & Sarker, 2009). However, leadership roles have not been fully investigated from any theoretical lens. In this study, we go beyond the proposed agile practices and examine the role of leadership and how leadership can provide an appropriate environment and support to help achieve agility. We study agility as a team capability instead of just using a specific agile method. We believe agility can be achieved by various ways in various contexts.

Leadership

Leadership has been recognized as an important factor in team success. Efforts have been made to study it from many different perspectives, including directive leadership that aims at structuring work by providing clear direction and expectations; transactional leadership where leaders will get what followers want to get the best performance from them; empowering leadership that promotes the sharing of power with subordinates in an attempt to raise their level of autonomy; and transformational leadership that attempts to inspire followers and influence them to contribute to the long-term benefits of the group to which they belong (Hoffman, Bynum, Piccolo, & Sutton, 2011; Lorinkova et al., 2013; Tyssena, Wald, & Spieth, 2014).

To answer our research questions, we adopted the leadership function perspective. The leadership function perspective focuses on leadership roles and the activities of each role. Prior research has identified several broad leadership roles in various group contexts. For example, the navigator role establishes and enacts the strategic vision among the members of a team and maintains a clear purpose and direction for the team (Carson, Tesluk, & Marrone, 2007; Contractor, DeChurch, Carson, Carter, & Keegan, 2012). The innovator challenges team assumptions and leads new initiatives, while the director establishes standards, negotiates tasks, and ensures deadlines (Klein, Ziegert, Knight, & Xiao, 2006; Zaccaroa, Rittmana, & Marks, 2001). The boundary spanner coordinates with outside stakeholders and maintains productive relationships with key external stakeholders, while the facilitator structures within-team coordination and tasks (Bonner, 2010; Carson et al., 2007; Contractor et al., 2012). The Social Integrator maintains healthy and productive social interactions, while the Engineer role structures the collective and the task, and coordinates member contributions (Carson et al., 2007). Most of this theoretical work was done in a general management context, not agile software development. In addition, detailed leadership activities are also proposed in the management literature, such as controlling timing, maintaining procedures, coaching, structuring activities, inspiring teams, and managing resources, as well as searching and structuring information, developing team members, identifying problems and requirements, and planning and providing feedback (Klein et al., 2006; Zaccaroa et al., 2001).

Among software development literature, the specific roles played by the Scrum Master and Product Owner have been outlined mainly in the practitioner literature. Moe et al. (2010) revealed many challenges in implementing agile methods and points out the importance of building an effective self-managing team from the beginning, but does not propose how. Bonner (2010) emphasized an individual's propensity for leadership success in an agile software development environment. Tessem (2014) analyzed how general agile practices (not leadership roles and behaviors) can empower teams. Faraj and Sambamurthy (2006) studied the difference between directive leadership and empowering leadership in non-agile software development where directive leadership behaviors include goal assignment, reprimand, and instructions, while empowering leadership behaviors include encouragement of teamwork and self-development, and participative goal setting.

There is a lack of studies on leadership functions (i.e., leadership roles and activities) in the context of agile software development. In this study, we will focus on leadership functions and investigate how various leader functions can help a team to achieve agility and better performance.

RESEARCH METHODOLOGY

This study adopts a case study research methodology. The nature of our study is exploratory, focusing on multifaceted concepts of leadership and their impact on a team's process and performance. Our aim is to generate theory that can describe leadership functions in agile methodologies. In such cases, qualitative data can better help understand the dynamic relationships among team process, agile methodologies, and leader behaviors (Goh et al., 2013). It enables us to develop a theoretical model to describe a leader's roles in various environments.

Data Collection

We intend to conduct our research using multiple cases, and are using replication sampling strategies to select cases (Eisenhardt, 1989; Eisenhardt & Graebner, 2007). We select companies based on their use and experience using agile methods. The criteria to select cases and participants are:

- The company has several years of experience using agile methods
- The methods have been widely adopted within the company
- The participants have several years of experience using the agile methods adopted by the company
- The participants play active roles in agile development in the company

We conducted three interviews at our first site, which is an application development division of a Fortune 500 financial institution. The division mainly develops internal financial systems. It started using Scrum in 2013. We interviewed one

agile coach and one Scrum master. The third interviewee first served as a Product Owner and recently transitioned to the role of Agile Coach.

We are in the process of collecting data at the second site, a large IT company that provides data and network services. The R&D division started using the SAFE framework two years ago. Prior to that, it used Scrum. We have interviewed two Product Owners and one Scrum Master. We are contacting other participants, including other Scrum Masters and Product Owners, as well as developers at this company.

Each interview lasted about 60 minutes. They were all recorded and transcribed. In addition to these two sites, we plan to contact two other organizations according to our site selection criteria. We plan to interview Product Owners, Scrum Masters, agile coaches and developers.

We prepared an interview checklist to guide our semi-structured interviews. The checklist was derived from our literature review and theoretical lens. It includes topics such as the interviewee's background, the company's background, the products being developed, agile methods used, challenges the team has faced over time, and the strategies the leader uses at different stages of the process. As we collected data, we updated and tailored the interview guide to solicit more information on emerging themes.

Data Analysis

We performed data analysis concurrently with data collection activities. Figure 1 describes the data analysis process we are following. Our analysis procedure consists of within-case data analysis followed by cross-case data analysis. Currently we are analyzing the data collected from the two sites. We followed grounded theory coding strategies suggested by Corbin & Strauss (2007) that includes open, axial, and selective coding.

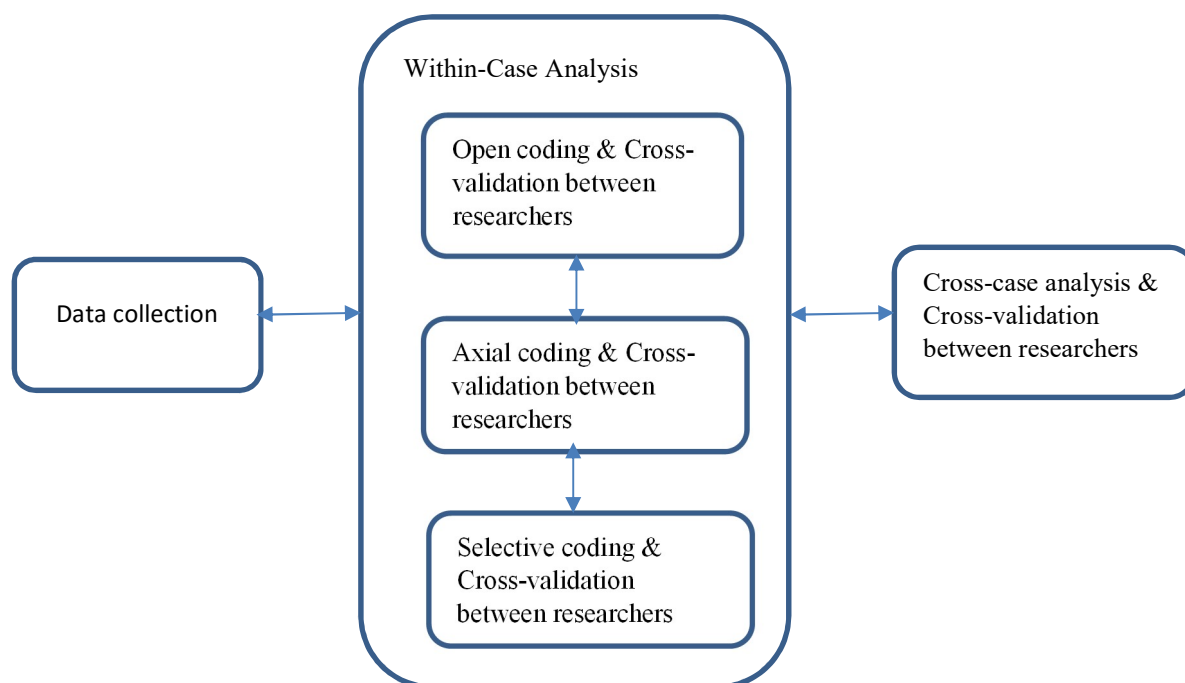


Figure 1. Data Analysis Process

PRELIMINARY FINDINGS

The current data suggests some preliminary findings. Figure 2 describes the preliminary research framework derived from our empirical data.

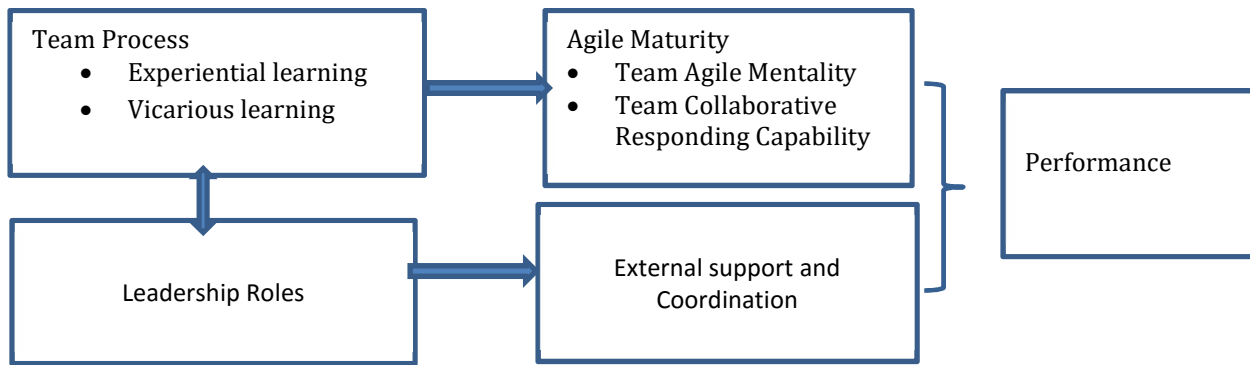


Figure 2. Preliminary Research Framework

Prior literature has described the Scrum Master’s role as clearing obstacles, addressing team dynamics, protecting the team from outside interruptions, and establishing the team environment, facilitating meetings, and ensuring agile rules in general (Bass, 2014; Dönmez, Grote, & Brusoni, 2016). The Product Owner’s role is described as representing the voice of the customer, collecting and prioritizing customer needs, and managing the product backlog (Dönmez et al., 2016; Vlietland et al., 2016). During our data analysis, the learning process emerged as a key process to lead to better agile performance. As a result, we identified several new leadership roles and corresponding activities that can help a team improve its learning processes, which in turn helps the build team an agile mentality and collaborative response capability. During the development process, a leader needs to play various roles, which include various activities, to support the team internally and externally. We discuss these roles below and summarize activities involved in each role in Table 1.

Observer

A good leader of an agile team needs to be a good observer. He or she needs to constantly scan the team’s environment and interactions, assess its direction, identify impediments and new opportunities, and develop intervention strategies.

Navigator

The navigator leadership role establishes and enacts a vision among members of a team and maintains a clear purpose and direction for the team. Leaders need to set product goals (e.g., features and timeline) and process goals (e.g., achieve higher team autonomy). This is especially crucial at the beginning of the project when team members are not clear about the project’s direction. During the development process, the leader needs to constantly remind the team about these goals. As a navigator, a leader also needs to help set and monitor the project’s boundaries for the team. The boundaries include both formal boundaries set by the company and informal team norms that everyone is expected to follow. For example, the leaders need to ensure that the team complies with regulatory standards set by the company; team members need to participate in the daily standup meetings; and the team is expected to use certain tools for reporting. The leader also needs to help the team negotiate the team norms (e.g., daily standup meeting time, dos and don’ts, etc.).

Initiator

The initiator role enacts and introduces new tasks, tools, processes and ideas to the team, especially at the beginning. This leader role aims to help shape the task structures, prioritize goals, formulate strategies and tactics, and explore innovative solutions. In the financial institution, the leaders help the team start the project by organizing the kickoff meeting, establishing agile routines, and prioritizing product goals. Recently, they introduced Kanban to the team as part of some new agile practices and led the team in learning how to effectively incorporate it into their current Scrum practices. During the meetings (e.g., planning meetings and retrospective meetings), the leader may need to initiate specific topics that he thinks the team needs to address immediately. Agile leaders have also introduced the agile forum where “agile stars” can share their experiences and recommendations with all the teams.

Cultivator

The cultivator role aims to build and maintain a productive, safe interactive environment where the team can autonomously function, improve their agile processes, collaborate among themselves, and innovate. The goal is to build an inspiring and empowering atmosphere so that team members can thrive. Leader activities in this role include building an ownership mentality, structuring a relational space where team members feel safe to innovate and experiment, facilitating the development of a conflict resolution agreement, and challenging team assumptions and non-agile practices. For example,

after identifying a bottleneck, a Scrum master continuously encouraged a team to brainstorm solutions and collectively make a decision. He also kept challenging the team's assumptions and solutions to intellectually stimulate the team. Another Scrum master helped the team recognize that trial-and-error is a part of the learning process and encouraged members to embrace it. An agile coach facilitated the sessions where the team developed a protocol regarding issue escalation.

External Coordinator

Leaders also play an external coordinator role managing the team's external environment and gaining support for the team's work. Leadership activities include developing social capital with the team's external stakeholders, such as middle level managers and the leaders of collaborating teams. Though there are formal communication channels among team leaders and between team leaders and middle level managers, a lot of communication and collaboration takes place via informal channels. Building relational capital with others can expedite the process. Other leader activities include managing dependency with another team, and escalating problems that the team cannot solve by itself (e.g., budget and deliveries).

Leader Role	Description	Leadership activities
Observer	Assess challenges and impediments, as well as identify opportunities to improve	<ul style="list-style-type: none"> Assess team environment Identify impediments and new opportunities for improvement
Navigator	Maintain a clear purpose and direction for the team	<ul style="list-style-type: none"> Set vision and direction Set and form team boundaries Ensure agile practices
Initiator	Initiate new tasks, tools, and practices	<ul style="list-style-type: none"> Initiate new practices Identify discussion topics Build agile task structure
External Facilitator	Manage the external environment of the team	<ul style="list-style-type: none"> Develop social capital with team stakeholders Manage team interdependence Escalate issues
Cultivator	Build and maintain healthy and productive social interactions within the team	<ul style="list-style-type: none"> Build ownership mentality Structure relational space for innovation Facilitate development of conflict resolution agreement Stimulate intellectual exploration Monitor hands-on treatment of agile practices

Table 1. Summary of Leader Roles and Activities

Team Learning Process

Our data suggests that the leadership roles and activities support two learning processes that help foster a team's agile mentality and capability to correspond collaboratively, which in turn leads to better agile performance. The two learning processes are experiential learning and vicarious learning. Experiential learning process focuses on learning by doing (Peng, Mu, & Benedetto, 2013). This process involves a team's exploration, experimentation, and reflection on what the team has done and what could be done differently (Bresman, 2013). For example, a team in our study was encouraged by its leader to reflect on their mistakes and discuss what could be done to achieve better results. The idea proposed by the team itself was explored and tested. Regardless of the results, such an exploration spirit was recognized and encouraged. By learning from their own decisions and actions, the team developed a better understanding of the project and process. In vicarious learning, the team learns from experts such as agile coaches, Scrum masters, and experienced teams (Peng, Mu, & Benedetto, 2013). During this process, the team forms its process and routine by absorbing a range of knowledge and suggestions from others (Bresman, 2013).

Our data suggest that the leadership roles and their activities can support and facilitate these two learning processes. The observer role helps identify impediments and improvement opportunities. It sets the stage for the learning process to begin. The navigator role sets clear direction and boundaries for the experiential learning process to take place. It ensures that although the team can explore its own strategies, it will stay on track. It helps maximize progress in the right direction, contributing to team effectiveness and performance. The initiator role enacts the task and process structures and helps kick off the team's two learning processes. Meanwhile, the cultivator creates a safe team environment where they can learn and enforce appropriate agile strategies either from their own experiments and/or through others' sharing. The external coordinator role helps develop a supportive external environment for the team to function.

Our data further suggest that the two learning processes that occur within a team can foster the team's agile maturity and develop its agile mentality. With such a mentality, the team can own its decisions, self-manage itself efficiently, and adjust its processes for the best results. It will also develop the team's capabilities to effectively and efficiently respond to the changes needed for quality delivery. An agile mentality and the capability to respond collaboratively, while supported by external support and coordination, will lead to high performance. The two learning processes will also provide feedback to leaders who can adjust their roles and activities accordingly. For example, a Scrum Master reported that he was more proactive at the beginning and actively involved in the team's meetings and decisions. With the growth of the team's agile maturity, his role became more passive and evolved into an observer most of the time.

CONCLUSION AND FUTURE WORK

In this study, we attempt to explore effective leadership functions in building agile teams. In this paper, we presented our preliminary findings. We are analyzing the data we have collected and are in the process of contacting more organizations and participants to obtain richer information. It will contribute to leadership theories in software development. Prior research on agile software development has concentrated on understanding agile practices from industry (Drury, Conboy, & Power, 2012), agile method adaptation (Ramesh, Mohan, & Cao, 2012), and agile practices (not leadership) that can empower team members (Tessem, 2014), or on comparing leadership styles in non-agile contexts (Faraj & Sambamurthy, 2006). High-level job descriptions of roles such as Scrum Master and Product Owner are available (Bass, 2014; Dönmez, Grote, & Brusoni, 2016; Vlietland et al., 2016), but they fail to address how an agile leader can achieve his or her goals effectively. There is also lack of systematic, theory-driven research in studying how effective leadership roles can help reshape a team and help the team achieve better performance. Our study will contribute to leadership theory in the context of agile software development. It will also add to our theoretical understanding of leadership and its process in team building. It will also make contributions to the IS profession because leadership roles and activities identified in this study can provide behavior guidelines to team leaders. The main challenge of this study is to continue to identifying relevant study sites that are willing to participate in our study. This study also has limitations. As an exploratory study, the results rely on qualitative data from a few study sites. Therefore, we need to be cautious when generalizing the findings to other settings.

REFERENCES

- Bass, J. M. (2014) Scrum master activities: Process tailoring in large enterprise projects, *Proceedings of the 2014 IEEE 9th International Conference on Global Software Engineering*, August 18-21, Washington, DC, USA.
- Bresman, H. (2013) Changing routines: A process model of vicarious group learning in pharmaceutical R&D, *Academy of Management Journal*, 56,1, 35-61.
- Bonner, N. A. (2010) Predicting leadership success in agile environments: An inquiring systems approach, *Academy of Information and Management Sciences Journal*, 13, 2, 83-103.
- Carson, J. B., & Tesluk, P. E. (2007) Leadership from within: A look at leadership roles in teams, *Proceedings of the 67th Annual Meeting of the Academy of Management*, Philadelphia.
- Carson, J. B., Tesluk, P. E., & Marrone, J. A. (2007) Shared leadership in teams: An investigation of antecedent conditions and performance, *Academy of Management Journal*, 50, 5, 1217-1234.
- Conboy, K. (2009) Agility from first principles: Reconstructing the concept of agility in information systems development, *Information Systems Research*, 20, 3, 329-354.
- Conboy, K., Coyle, S., Wang, X., & Pikkarainen, M. (2011) People over process: Key challenges in agile development, *IEEE Software*, 28, 4, 48-57.
- Contractor, N. S., DeChurch, L. A., Carson, J., Carter, D. R., & Keegan, B. (2012) The topology of collective leadership, *The Leadership Quarterly*, 23, 6, 994-1011.
- Corbin, J., & Strauss, A. (2007) Basics of qualitative research: Techniques and procedures for developing grounded theory (3rd ed.), SAGE Publications, Los Angeles.
- Dönmez, D., Grote, G., & Brusoni, S. (2016) Routine interdependencies as a source of stability and flexibility: A study of agile software development teams, *Information and Organization*, 26, 63-83.

- 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.
- Eisenhardt, K. M. (1989) Building Theories from case study research, *Academic Management Review*, 14, 4, 532-550.
- Eisenhardt, K. M., & Graebner, M. E. (2007) Theory building from cases: Opportunities and challenges. *Academy of Management Journal*, 50, 1, 25-32.
- Faraj, S., & Sambamurthy, V. (2006) Leadership of information systems development projects, *IEEE Transactions on Engineering Management*, 53, 2, 238-249.
- Goh, J. C.-L., Pan, S. L., & Zuo, M. (2013) Developing the agile IS development practices in large-scale IT projects: The trust-mediated organizational controls and IT project team capabilities perspectives, *Journal of the Association for Information Systems*, 14, 12, 722-756.
- Gren, L., Torkar, R., & Feldt, R. (2017) Group development and group maturity when building agile teams: A qualitative and quantitative investigation at eight large companies, *The Journal of Systems and Software*, 124, 104-119.
- Hoffman, B. J., Bynum, B. H., Piccolo, R. F., & Sutton, A. W. (2011) Person-organization value congruence: How transformational leaders influence work group effectiveness, *Academy of Management Journal*, 54, 4, 779-796.
- Klein, K. J., Ziegert, J. C., Knight, A. P., & Xiao, Y. (2006) Dynamic delegation: Shared, hierarchical, and deindividualized leadership in extreme action teams, *Administrative Science Quarterly*, 51, 590-621.
- Lee, G., & Xia, W. (2010) Toward agile: An integrated analysis of quantitative and qualitative field data, *MIS Quarterly*, 34, 1, 87-114.
- Lindstrom, L., & Jeffries, R. (2004) Extreme programming and agile software development methodologies, *Information Systems Management*, 21, 3, 41-52.
- Lorinkova, N. M., Pearsall, M. J., & Sims, H. P. J. (2013) Examining the differential longitudinal performance of directive versus empowering leadership in teams, *Academy of Management Journal*, 56, 2, 573-596.
- Magni, M., & Maruping, L. M. (2013) Sink or swim: Empowering leadership and overload in teams' ability to deal with the unexpected, *Human Resource Management*, 52, 5, 715-739.
- 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.
- Meso, P., & Jain, R. (2006) Agile software development: Adaptive systems principles and best practices, *Information Systems Management*, 23, 3, 19-30.
- Moe, N. B., Dingsøyr, T., & Dybå, T. (2010) A teamwork model for understanding an agile team: A case study of a Scrum project, *Information and Software Technology*, 52, 480-491.
- Müller, R., Sankaran, S., Drouin, N., Vaagaasar, A.-L., Bekker, M. C., & Jain, K. (2018) A theory framework for balancing vertical and horizontal leadership in projects, *International Journal of Project Management*, 36, 1, 83-94.
- Peng, G., Mu, J., & Benedetto, C. A. D. (2013). Learning and open source software license choice, *Decision Sciences*, 44, 4, 619-643.
- Ramesh, B., Mohan, K., & Cao, L. (2012) Ambidexterity in agile distributed development: An empirical investigation, *Information Systems Research*, 23, 2, 323-339.
- Sarker, S., Munson, C. L., Sarker, S., & Chakraborty, S. (2009) Assessing the relative contribution of the facets of agility to distributed systems development success: An analytic hierarchy process approach, *European Journal of Information Systems*, 18, 285-299.
- 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.
- Tessem, B. (2014) Individual empowerment of agile and non-agile software developers in small teams, *Information and Software Technology*, 56, 8, 873-889.
- Tyssena, A. K., Wald, A., & Spieth, P. (2014) The challenge of transactional and transformational leadership in projects, *International Journal of Project Management*, 32, 3, 365-375.
- Vinekar, V., Slinkman, C. W., & Nerur, S. (2006) Can agile and traditional systems development approaches coexist? An ambidextrous view, *Information Systems Management*, 23, 3, 31-42.
- Vlietland, J., Solingen, v., & van Vliet, H. (2016) Aligning codependent Scrum teams to enable fast business value delivery: A governance framework and set of intervention actions, *Journal of Systems & Software*, 13, 418-429.
- Yu, X., & Petter, S. (2014) Understanding agile software development practices using shared mental models theory, *Information and Software Technology*, 56, 911-921.
- Zaccaroa, S. J., Rittmana, A. L., & Marks, M. A. (2001) Team leadership, *The Leadership Quarterly*, 12, 4, 451-483.