

Association for Information Systems

AIS Electronic Library (AISeL)

International Research Workshop on IT Project
Management 2019

International Research Workshop on IT Project
Management (IRWITPM)

12-14-2019

Transforming Management to Support Agile Development

Peng Xu

Yide Shen

Follow this and additional works at: <https://aisel.aisnet.org/irwitpm2019>

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 2019 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Transforming Management to Support Agile 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

Agile methods have been widely adopted in the software development industry. Efforts have been made to study how to adopt and tailor agile methodologies for various types of projects. However, studies have reported that many of the challenges that today's agile teams face do not come from within their teams, instead, they mainly come from the interaction between agile teams and their business environment. Thus, being agile is not just a task for development teams; it needs to go beyond the team level and requires transformation at the management level. This study aims to investigate how management external to agile teams needs to be redefined and transformed in order to fully support agile development.

Keywords

Agile development, management transformation, leadership capability

INTRODUCTION

Over the years, software development methodologies have evolved from ad hoc approaches to a proliferation of methodologies. Agile methodologies are one approach that has been widely adopted in the software development industry. Though various agile methodologies differ in practices, tools, and many other features, they all share common agile principles such as satisfying customers as the highest priority, embracing changing requirements, focusing on working software, frequent delivery, and frequent communications¹. Different from traditional “heavyweight” methodologies, agile methodologies rely on dynamic adaptation and lean processes (Maruping, Venkatesh, and Agarwal, 2009; Sarker and Sarker, 2009).

Efforts have been made to study how to adopt and tailor agile methodologies for various types of projects. Prior research has focused much of its attention on agile practices, their suitability and implementation at the team level (Ramesh, Mohan, and Cao, 2012; Sarker and Sarker, 2009). Such research has facilitated agile adoption by various teams and projects. Businesses have realized that the goal of adopting agile methods is to help teams become more flexible and open to the changes posed within and/or outside the company. While plenty of research has studied how to leverage various agile practices and achieve agility at the team level, the challenges of lacking management support have not drawn sufficient attention. Most research on agile methods concentrates on agile practice adoption at the team level and assumes that agile methods only have an impact on the development team. However, studies have reported that many of today's challenges do not come within the team, but instead they mainly come from the interaction between an agile team and its business environment, especially from the management and business stakeholders that interface with development teams directly (Gregory, Barroca, Sharp, Deshpande, and Taylor, 2016). A survey aiming to identify the main challenges in current agile development found that many teams struggled to identify how to communicate their teams' processes to management and to identify what needs to be reported to management (Gregory et al., 2016). In addition, agility of development teams is also affected by management teams who still tend to rely on control-and-demand mindset, resist culture changes to support agile development, keep old bureaucracy and silos, and follow the traditional water-fall management strategies when plan and monitor deadlines, processes, and resources (Dikert, Paasivaara, and Lassenius, 2016). Such management may pose problems to the effectiveness of agile development. This issue is more problematic in large organizations due to its rigid, hierarchical structures and culture.

¹ <https://www.agilealliance.org/agile101/12-principles-behind-the-agile-manifesto/>

All of these challenges suggest that being agile is not just a task for development teams. It needs to go beyond adopting agile practices at the team level. It requires transformation of the management external to development teams and calls for a new style of management and capabilities. In this study, the management external to development teams refers to the IT and business managers who and/or whose units interface with development teams and have direct influence on teams and software products. Such management includes project management office, product management office, agile methodology sponsors, and business functional managers (e.g., marketing, sales) whose decisions can affect software products developed by teams. Such leadership is crucial for team effectiveness (Carson and Tesluk, 2007; Lorinkova, Pearsall, and Sims, 2013). It can influence a project's objectives, motivate desired behaviors in pursuit of these objectives, change business processes, and affect group culture (Carson and Tesluk, 2007). An agile team that operates in a rigid, control-and-command environment cannot fully leverage the benefits of being agile and cannot act quickly to respond to changes demanded by the market and/or customers. However, we know surprisingly little about how management external to development teams should reposition their roles and re-define their capabilities to support agility. Motivated by such a research gap, this study aims to investigate how management outside of agile teams needs to be transformed in order to support agile teams to achieve agility.

In the following sections, we discuss the theoretical background and describe our research methodology. We conclude with potential contributions to theory and practice.

THEORETICAL BACKGROUND

Challenges in Achieving Software Development Agility

Emerging during the 1990s as a software development methodology, agile has gathered momentum as a more collaborative and responsive approach than the traditional waterfall approach (Birkinshaw, 2019). The reported benefits of agile methods include increased productivity, faster turnaround, and higher developer satisfaction (Lindstrom and Jeffries, 2004). To achieve software development agility, various agile methodologies such as Scrum have been proposed (Lee and Xia, 2010; Moe, Dingsøyr, and Dybå, 2010). These agile methodologies propose practices that a software team should follow, such as short iterations, daily meetings, frequent releases, minimal planning, and working products among others. Although various agile methods differ in techniques, processes, tools, and other features, they all share common principles that values such as satisfying customers as the highest priority, frequent delivery, embracing changing requirements, focusing on working software, and frequent communications. Agile teams are encouraged to self-manage, instead of being micromanaged (Gren, Torkar, and Feldt, 2017).

While the understanding of agile practices and how to achieve agility within an agile team is maturing, research has found that what prevent agile teams from achieving their agility goals does not merely come from the team, but also from the team's environment. A recent agile report indicates that internal organizational culture remains as one of the main obstacles for success in many organizations because some organizational culture is at odds with agile values². This is not surprising since an agile driven process is likely to change, if not diminish, the power balance within management.

One challenge agile teams face is how their work is managed by managers who do not yet fully understand agile values and principles. This challenge stems from the conflict between the traditional command-and-control role of a manager and agile's leading and coaching role (Yi, 2011). In some organizations, project managers naturally become agile team leaders without fully understand agile methods and values. As a result, they hold onto the old habit of dictating how and how much work should be done within an iteration (Long and Starr, 2008). A similar situation can also happen in cases where managers make developers felt micromanaged and create tension between the manager and developers (Lee, 2008). The other challenge for agile teams is how to communicate with management and non-agile business units when they have different perceptions about project success, communication standards, and change management. While development teams are using agile practices, their projects are still being approved, budgeted and monitored by management that hasn't yet adopted an agile mindset and practices (Gregory et al., 2016). Management still values formal and written communication, expects certainty of time, budget, and specifications from the start of the project, and perceives re-prioritization and cutting project scope as lack of control. As a result, agile teams struggle to figure out what and how to communicate with management. Such struggle also exists when agile teams need to communicate and collaborated non-agile business units (Gregory et al., 2016; Long and Starr 2008).

² <https://explore.versionone.com/state-of-agile>

The above challenges show that in many organizations, managers are unwilling or unable to change the way they interact with agile development teams. Many managers still have a tendency to revert to old way of working when having impediments; they still rely heavily on old bureaucracy to control and police development progress; and old commitments such as firm deadlines are still expected to be kept (Dikert et al., 2016; Long and Starr, 2008; Yi, 2011). However, the agile maturing process requires a transformation at the management level (Dikert et al., 2016). Due to the important role that management plays in any organizational change, it is vital for managers to change their mindset, attitudes and roles when working in an agile environment (Abdelnour-Nocera and Sharp, 2008; Cowan, 2011). In doing so, they need to be involved in fixing the lines of authority, understanding what agile practices mean to every process for all stakeholders, and transforming management through the change process (Abdelnour-Nocera and Sharp, 2008; Cowan, 2011). Development teams cannot leverage the full benefits of agile development unless the interfaced management teams and non-agile units also embrace the same agile values and work along the same agile paradigm. Otherwise, it is likely to undermine agile projects, either intentionally or out of ignorance (Spayd, 2003). As stated by Yi (2011), “Without changing the organizational management, self-managing teams are more accidental” (p. 153). Thus, we need to further understand how management needs to transform itself to fit with agile development and support agile development

Management Transformation and Agile Principles

Existing literature on management transformation and agile practices focuses on overall organizational structure changes and re-design when agile principles are introduced to management. Although agile principles first emerged as a responsive and collaborative approach to software development, they have started moving into mainstream thinking as a management practice (Birkinshaw, 2019). For example, a financial institution restructured its operations in an European country borrowing agile principles and values (Birkinshaw, 2019). It adopted a “big bang” approach with dramatic changes affecting all aspects of the business. The changes include major changes to job descriptions, a significant downsizing in the organization, and re-organizing people into squads and tribes around customers, etc. While some have attempted to change organizational structure to support agility, others point out that restructuring is not the answer because organizational structures can create silos and trap employees. Instead, businesses need to “lead change by assigning accountabilities” to self-managing units to achieve agility (Ross, 2019). The challenges in such transformations are that the agile practices derived from agile principles are directly associated with software development tasks. Thus, initial adoption of agile practices by software development teams can rely on the guidance of specific agile method descriptions. However, when implementing agile principles outside the pre-defined software domain, one needs to modify the practices in ways that “both match the new context and preserve the underlying principles” (Repenning, Kieffer, and Repenning, 2019).

These studies represent the beginning of efforts to understand how agile methods and agile values affect overall management strategies. The focus of such studies is on overall organizational structure changes when applying agile principles to organizations. They do not address the challenges of how management should redefine its roles and transform itself to support agile development. To fill this research gap, we propose to use Complex Adaptive Systems (CAS) theory to study management transformation process to support agile development.

Complex Adaptive Systems Theory

Complex Adaptive Systems (CAS) theory describes how individual parts of a system interact with one another and shape the whole system’s behaviors (Holland, 1995). It captures the various entities involved, their actions, and their relationships with their surroundings in an adaptive system. Complex Adaptive Systems theory includes three major components: agent, interaction, and environment (Holland, 1995). Agents are individual actors in a complex adaptive system. Agents can be various entities such as human beings, objects, and even organizations. Each agent has its own attributes and behavior rules. Attributes define an agent’s properties while behavior rules guide the agent’s actions and processes in the system. Both attributes and behavior rules can change to respond to feedback provided by adaptive systems.

Interactions in CAS describe adaptive behaviors of agents affected by each other. Such interactions can also take many forms based on the adaptive system in a study. The third component in CAS, environment, refers to the context where agents interact with one another to complete certain actions and achieve certain goals. An environment in CAS can be defined by structures which characterize the topography of the environment. The structure in an environment defines the conditions and boundaries of actions.

The three components of CAS affect each other during adaptive processes. For example, feedback from its environment and other agents can change the attributes and behavioral rules of an agent. Such new attributes and rules can affect interactions with others within the systems. Meanwhile, environmental structures can also adapt to new demands of agents and their interactions.

Complex Adaptive Systems theory has been applied to multiple domains, including IT (Nan, 2011; Orlikowski, 1996). Orlikowski (1996) examined an organizational transformation caused by introducing new information systems. Agents in this case are human actors and IT features. Interactions refers to IT use processes between human actors and IT features and among human actors. The environment is the social or organizational contexts of IT use processes (Nan, 2011). Orlikowski's (1996) study shows how transformation is enacted through the situated practices of IT users as they improvise, innovate, and adjust their work routines over time while adopting an information system. It shows that both IT use and organization transformation are the products of self-orchestrated interactions among users, technologies, and organizational environment. Such a transformation cannot just be dictated by organization command (Nan, 2011).

Prior research on IT use has demonstrated that CAS theory is a good fit to study the process of transformation. Figure 1 shows the research framework derived from CAS theory for this study. Our research goal is to understand how management external to agile development teams should be redefined and transformed in order to fully support agile development. Agents in this study include human actors such as management, agile teams, and agile leaders and non-human IT product. The arrows in the figure represent interactions among various agents. Interactions in this context refers to collaborations within agile teams and between agile teams, and cooperation between agile teams and management. Contextual structure defines environments where agents interact with one another. Such contextual structures include both corporate structures, organizational norms and policy, and the structures of agile teams. We believe interactions among these components and feedback provided by adaptive systems will lead to new patterns of management in an agile organization. Thus, we will use CAS theory as our fundamental theory base to guide our study, research design, and data analysis.

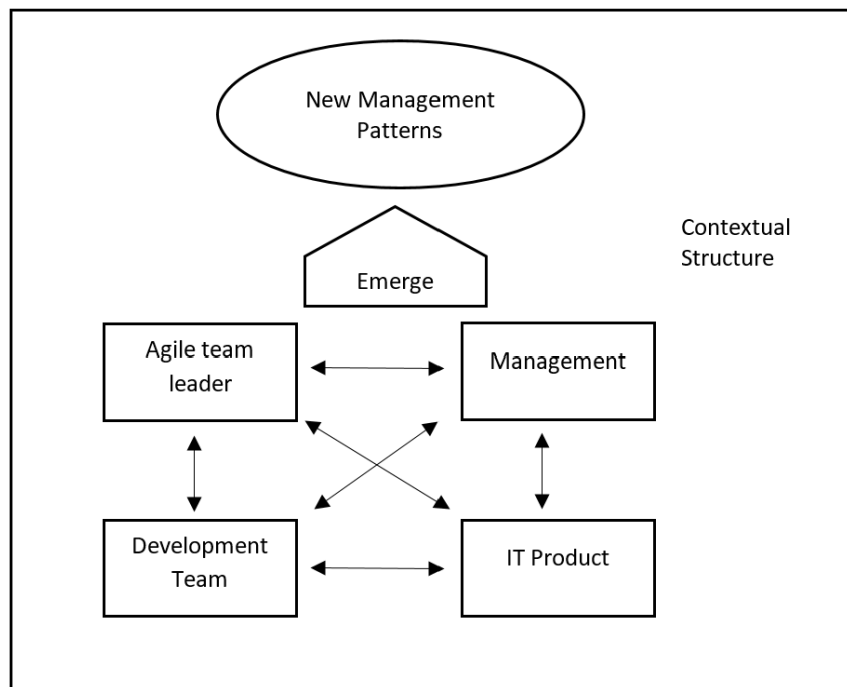


Figure 1. Research Framework

RESEARCH METHODOLOGY

We plan to use a case study research methodology to develop a theory that can explain the management changes in an organization that has adopted agile methods. The nature of our study is exploratory. Our aim is to generate theory that can describe how management external to agile teams needs to be re-defined and transformed in order to fully support agile development. In such cases, qualitative data can better help understand the dynamic relationships among various parts of an organization that interact with one another (Goh et al., 2013). We only focus on understanding management transformation for co-located agile development teams in large organizations in this study. It enables us to develop a theoretical model to describe the transformation process.

Data Collection

We intend to conduct our research using multiple cases, and employing replication sampling strategies to select cases (Eisenhardt and Graebner, 2007). We will select large organizations based on their use and experience with agile methods. We plan to identify two sites. They should have successfully used agile methods company-wide. They also should have started transforming their management and seen promising results. We plan to rely on interviews to collect qualitative data. All interviews will be recorded, if permitted, and transcribed. Otherwise, we will take detailed notes. We plan to interview participants spanning vertical levels affected by agile transformation. The potential candidates include agile leaders, agile coaches, product owners, product/project managers, middle management from software development, business stakeholders, and even senior managers if they are also involved and/or affected by agile processes. The initial interview questions will use open-ended questions to solicit the history of management changes in the organization. Questions will focus on (a) how the organization first realized the challenges in management when adopting agile methods at the team level; (b) what challenges they were facing; (c) what triggered the change; (d) what strategies were used; (e) how changes were made, (f) what were the impact of these changes, etc. We will also seek other ways to obtain data such as document reviews and observation, if possible.

Data Analysis

We will do within-case data analysis first followed by cross-case data analysis. We plan to follow grounded theory coding strategies suggested by Corbin and Strauss (2007) that includes open, axial, and selective coding. In the open coding process, we plan to read and analyze all transcripts separately. After identifying common themes and patterns, we will discuss these common concepts. Then, we will re-analyze all transcripts with the common concepts. The process of open coding will be iterative. When we develop and agree with a set of common concepts, we will move to axial coding where we will explore the relationships among common concepts identified in open coding. Again, we will do it separately first and discuss our findings with one another. After understanding the relationships, we will start the process of selective coding where we define the core concepts and develop the process model.

CONCLUSION AND FUTURE WORK

In this study, we aim to explore how to transform management to support agile development. Prior attention has been paid to agile practices at the team level. However, being agile is beyond making developers follow agile methods. The rigid environment and traditional managerial approach become major hurdles for agile teams to achieve their agility goals. Thus, it is crucial for management to re-define itself regarding its functions, styles, and capabilities to be able to support agility. It is interesting to see how agile methods lead to organizational transformation. This study has several potential contributions. First, it will contribute to the agile literature by presenting a research model that shows how management should effectively interface with agile development teams. Second, it will contribute to the organizational transformation literature and business agility literature. Business agility is deemed as a necessary capability in today's business environment. Research on agility has proposed what business capabilities an agile organization should have (Overby, Bharadwaj, and Sambamurthy, 2006; Sambamurthy, Bharadwaj, and Grover, 2003), but few have studied the impact of agile development on organizational level agility. Little is known about reorganization and restructuring triggered by agile development. The findings of this research will address this gap. This study can also contribute to practices. The findings on how to transform management in an agile environment can provide guidelines for organizations that are struggling with this challenge.

REFERENCES

Abdelnour-Nocera, J., and Sharp, H. (2008). *Adopting agile in a large organisation*. Paper presented at the Agile Processes in Software Engineering and Extreme Programming, June 10-14, Limerick, Ireland.

- Birkinshaw, J. (2019). What to Expect From Agile. *MIT Sloan Management Review, Special Collection, Staying Agile*, 8-11.
- Carson, J. B., and Tesluk, P. E. (2007). *Leadership from within: A look at leadership roles in teams*. Paper presented at the the 67th Annual Meeting of the Academy of Management, Philadelphia.
- Corbin, J., and Strauss, A. (2007). *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory* (3 ed.). Log Angeles: SAGE Publications.
- Cowan, C. L. (2011). *When the VP is a Scrum Master, You Hit the Ground Running*. Paper presented at the Agile Conference, August 8 – 12, Salt Lake City.
- Dikert, K., Paasivaara, M., and Lassenius, C. (2016). Challenges and Success Factors for Large-Scale Agile Transformations: A Systematic Literature Review. *The Journal of Systems and Software*, 119, 87–108.
- Eisenhardt, K. M., and Graebner, M. E. (2007). Theory Building From Cases: Opportunities And Challenges. *Academy of Management Journal*, 50,1, 25–32.
- Goh, J. C.-L., Pan, S. L., and 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.
- Gregory, P., Barroca, L., Sharp, H., Deshpande, A., and Taylor, K. (2016). The Challenges That Challenge: Engaging with agile practitioners' concerns. *Information and Software Technology*, 77, 92–104.
- Gren, L., Torkar, R., and 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.
- Holland, J. H. (1995). *Hidden Order: How Adaptation Builds Complexity*. Reading, MA: Perseus Books.
- Lee, E. C. (2008). *Forming to Performing: Transitioning Large-scale Project into Agile*. Paper presented at the Agile Conference, Augus 4-8, Toronto
- Lee, G., and Xia, W. (2010). Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data. *MIS Quarterly*, 34,1, 87-114.
- Lindstrom, L., and Jeffries, R. (2004). Extreme Programming And Agile Software Development Methodologies. *Information Systems Management*, 21,3, 41-52.
- Long, K., and Starr, D. (2008). *Agile Supports Improved Culture and Quality for Healthwise*. Paper presented at the Agile Conference, Augus 4-8, Toronto
- Lorinkova, N. M., Pearsall, M. J., and 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.
- Maruping, L. M., Venkatesh, V., and Agarwal, R. (2009). A Control Theory Perspective on Agile Methodology Use and Changing User Requirements. *Information Systems Research*, 20,3, 377–399.
- Moe, N. B., Dingsøyr, T., and 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.
- Nan, N. (2011). Capturing Bottom-Up Information Technology Use Processes: A Complex Adaptive Systems Model. *MIS Quarterly*, 35,2, 505-532.
- Orlikowski, W. J. (1996). Improvising Organizational Transformation Over Time: A Situated Change Perspective. *Information Systems Research*, 7,1, 63-92.
- Overby, E., Bharadwaj, A., and Sambamurthy, V. (2006). Enterprise Agility and the Enabling Role of Information Technology. *European Journal of Information Systems*, 15,2, 120-131.
- Ramesh, B., Mohan, K., and Cao, L. (2012). Ambidexterity in Agile Distributed Development: An Empirical Investigation. *Information Systems Research*, 23,2, 323-339.
- Repenning, N. P., Kieffer, D., and Repenning, J. (2019). A New Approach to Designing Work. *MIT Sloan Management Review, Special Collection "Staying Agile"*, 12-21.
- Ross, J. (2019). Goodbye Structure; Hello Accountability. *MIT Sloan Management Review, Special Collection, Staying Agile*, 22-25.
- Sambamurthy, V., Bharadwaj, A., and Grover, V. (2003). Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms. *MIS Quarterly*, 27,2, 237-263.
- Sarker, S., and 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.
- Spayd, M. K. (2003). *Evolving Agile in the Enterprise: Implementing XP on a Grand Scale*. Paper presented at the Agile Development Conference, Salt Lake City.
- Yi, L. (2011). *Manager as Scrum Master*. Paper presented at the Agile Conference, August 8 – 12, Salt Lake City.