Applied Human Factors and Ergonomics (AHFE2025), Vol. 199, 2025, 2429–2437
https://doi.org/10.54941/ahfe1007055

Effects of Shared Leadership and Absorptive Capacity on Software Development Agility: A Preliminary Study

Szu-Yu Chen[1], Jung-Chieh Lee[2], Chung-Yang Chen[1], and Pei-Chi Chen[3]

[1]Dept of Information Management, National Central University, Taiwan
[2]Beijing Normal University at Zhuhai, China
[3]Trend Micro Inc., Taiwan

ABSTRACT

In modern software development, teams face numerous challenges, particularly frequent and diverse changes in requirements. The ability to respond promptly and effectively to such changes depends not only on technical expertise but also on human and team factors—specifically, how members learn from one another and leverage their diverse backgrounds. To succeed, teams must cultivate software development agility (SDA) supported by effective leadership capable of guiding them through shifting requirements. Grounded in dynamic capability theory, this study investigates the relationship between shared leadership and absorptive capacity and their combined influence on software development agility. The proposed theoretical model illustrates how shared leadership enhances a team's absorptive capacity, thereby fostering SDA. This work contributes to the human-centered software development literature by offering new insights into team dynamics within the context of requirement changes, viewed through the lens of dynamic capability theory.

Keywords: Human-side software development, Requirements change, Software development agility, Shared leadership, Absorptive capacity

INTRODUCTION

In today's rapid and dynamic software development environments, software development teams face frequent requirements changes (Spagnoletti et al., 2021; Laplante and Kassab, 2022), and the capacity of software teams to effectively respond to changing requirements has emerged as a critical area of inquiry. The ability of a software team to swiftly adapt to changes in requirements during the development process is referred to as Software Development Agility (SDA) (Lee and Xia, 2010; Baham and Hirschheim, 2022; Rathor et al., 2024). A software team is expected to have such an ability to respond promptly to users' changing needs and to formulate comprehensive, adaptive solutions. Within this context, how to lead a team through changes to accomplish development agility is particularly important. Team leaders play a pivotal role in guiding software development

processes, particularly when teams face multifaceted challenges across different domains. These challenges often require leaders to possess diverse domain-specific expertise, solution design capabilities, and technological management skills (Chow and Cao, 2008).

Software development is a process that relies heavily on knowledge (Bjørnson and Dingsøyr, 2008), particularly due to the variety and novelty of requirements changes. Thus, software teams are also expected to effectively learn and leverage new knowledge to come up with innovative solutions to address changes (Chen and Lee, 2025). In literature, absorptive capacity, originzted in the oprganizational leagning domain, is recognized as a dynamic capability (Teece, 2012) that empowers teams to proactively seek out external knowledge sources and implement effective systems for acquiring, assimilating, transforming, and utilizing this external knowledge (Lee et al., 2021; Zollo and Winter, 2002). To software teams, possessing this capacity to absorb external knowledge is seemly an essential competency to navigate fluctuating demands and environmental uncertainty (Lee et al., 2018). Building on this premise, when software teams encounter shifting requirements, leadership, particularly the shared leadership enables members with relevant expertise to assume leadership roles, guiding the team toward a collective understanding of the changes and facilitating solution development.

With the above backdrop, we argue that given the dynamic and multifaceted nature of modern software development, shared leadership may act as a potentially valuable approach in the context of requirements changes. Unlike traditional single-leader models, shared leadership emphasizes distributed leadership processes, where both formal leaders and team members collaboratively engage in planning, dynamic coordination, and shared decision-making. This model fosters a greater openness to member input, strengthens collective vision, and enhances collaborative problem-solving (Wang et al., 2014; Cullen-Lester and Yammarino, 2016). Therefore, this study seeks to explore whether shared leadership enables teams to respond more effectively and swiftly to changing requirements, ultimately enhancing their SDA capabilities. By conceptualizing the interplay between shared leadership and SDA, this research aims to contribute to a deeper understanding of how leadership dynamics can be leveraged to improve adaptive capabilities within contemporary software development teams. To the research of the SDA, this paper is an interdisciplinary study that utilizes absorptive capacity from the organization learning domain to address team dynamics in requirements changes, and is a preliminary study that theorizes a model for future empirical investigation. The rest of this paper is organized as follows. Section 2 surveys relevant literature; Section 3 conceptualizes the model and its components; Section 4 discuss possible confounding factors to the model design. Section 5 outlines the future steps of the entire research.

LITERATURE SURVEY

- **Software Development Agility (SDA)**

Agility is essential for sustaining a competitive edge in dynamic business environments. In the context of software development, user needs frequently evolve and changes frequently occur. Agility plays a pivotal role not only in addressing shifting user needs but also in maintaining development coherence (Rathor, 2024; Maruping et al., 2009). Lee and Xia (2010) proposed a widely recognized two-dimensional framework for measuring software development agility, which consists of response extensiveness and response efficiency. Response extensiveness refers to the extent to which a software team can accommodate and address changes in the needs of various types of users throughout the project lifecycle. A broader response capacity enables the team to more comprehensively meet the diverse and evolving needs of stakeholders. Response efficiency focuses on minimizing the time, cost, manpower, and other resources expended in adapting to changes. High response efficiency indicates that a team can adjust to shifting requirements in a timely and cost-effective manner. By evaluating response extensiveness and efficiency, one can assess the degree to which a software team achieves SDA, specifically maintaining process efficiency while satisfying user needs (Baham and Hirschheim, 2022).

Since software requirement fluctuations are both frequent and intricate, to realize software development agility, teams must establish robust agile practices, particularly in the domains of communication and team-based decision-making (Rathor, 2024). A critical element of agility in empirical software development is the team's dynamic capability to collaboratively learn from changes and reflect the learning to refine the development. This continuous learning process enhances both the speed of response and the capability to adjust to future changes, ultimately improving the overall development workflow (Baham and Hirschheim, 2022).

- **Dynamic Capability Theory and Absorptive Capacity**

The dynamic capability theory articulated by Eisenhardt and Martin (2000) pertains to the collaborative processes within organizations that enable the creation, integration, and reconfiguration of resources to secure a competitive edge in response to environmental fluctuations. This capability empowers teams to establish routines for acquiring knowledge from external stakeholders, including customers, suppliers, and research institutions. The theory of dynamic capabilities offers a useful framework for understanding absorptive capacity, facilitating firms in actively seeking external knowledge sources and developing effective systems for identifying, absorbing, and leveraging this knowledge (Zollo and Winter, 2002) to enhance their competitiveness. Importantly, absorptive capacity is understood as both a manifestation and a foundational element of dynamic capabilities, particularly in relation to an organization's ability to sense and respond to changes (Lee et al., 2021; Teece, 2012).

Absorptive capacity should thus not be viewed merely as a static process of team learning; rather, it represents a dynamic capability that allows teams

to build enough flexibility to address evolving needs and navigate uncertain and fluctuating environments. The foundational absorptive capacity theory, initially articulated by Cohen and Levinthal (1990), defines absorptive capacity as a team's ability to recognize the value of external knowledge, assimilate it, and apply it effectively. This theory underscores the close relationship between a firm's absorptive capacity and its existing knowledge base, internal learning processes, and innovation potential.

- **Shared Leadership**

When software development teams face shifting requirements, they inevitably encounter a range of specialized and domain-specific challenges. This necessitates that leadership within the team adapt rapidly to address these professional complexities. Leadership is widely recognized as a critical driver of knowledge sharing and team learning (García-Morales et al., 2012; Ali et al., 2018). Shared leadership may offer a valuable mechanism for doing so, as it allows team members with domain-specific expertise to assume leadership roles in addressing issues (Langfred, 2000). By fostering these processes, shared leadership may play a vital role in enhancing a team's absorptive capacity-- its ability to assimilate and apply new knowledge, which, in turn, enables teams to effectively adapt to emerging technologies and evolving requirements.

According to the literature, shared leadership is an innovative approach to leadership that prioritizes a collective vision and collaborative action plans between leaders and team members, in contrast to conventional hierarchical leadership models (Strode et al., 2022; Xu, 2022; Kimura, 2024). Leaders who adopt this leadership promote the utilization of team members' expertise, actively listen to their viewpoints, foster psychological empowerment and intrinsic motivation, involve team members in the decision-making process, and share accountability, all of which contribute positively to team performance (Wang, 2014). In the context of modern software development teams, the increasing complexity and variability of work demands have amplified the need for diverse knowledge and skills. No single individual is likely to possess the full range of competencies required to effectively navigate the multifaceted challenges faced by such teams (Strode, 2022). As a result, the limitations of the traditional leader model have become increasingly apparent, underscoring the relevance and necessity of adopting shared leadership approaches to effectively meet the evolving demands and complexities inherent in contemporary software development environments.

CONCEPT DEVELOPMENT

- **The Influence of Shared Leadership on Absorptive Capacity**

Gronn (2002) conceptualizes shared leadership as a "dynamic and fluid" process in which leadership responsibilities are allocated among team members according to situational demands, allowing leadership authority and responsibilities to shift flexibly among members as task requirements evolve. Within software development teams, shared leadership facilitates

the transfer of leadership roles to individuals who possess the critical knowledge, skills, and competencies needed to address specific challenges facing the team at any given moment (Langfred, 2000). Rather than relying solely on a designated supervisor, decision-making authority is distributed among domain experts within the team, enabling each member to actively participate in the decision-making process based on their expertise and experience, thereby contributing to the achievement of team goals. By assigning leadership roles to members with specialized knowledge and supporting them through collaborative and complementary efforts from other members, shared leadership enhances the flow of knowledge within the team. This process enables the team to develop a deeper understanding of problems, address highly variable requirements, and absorb relevant new knowledge, ultimately strengthening the team's absorptive capacity and improving overall performance (Hoch and Dulebohn, 2017).

Moreover, since shared leadership empowers team members to make decisions based on their expertise and the situational context (Wang, 2014), it reduces the delays typically associated with hierarchical reporting and approval processes under traditional leadership structures, thereby enhancing response efficiency and promoting the seamless flow of information within the team. As a result, all team members can access the necessary knowledge in real time, enabling them to make informed adjustments and decisions. This dynamic not only improves the team's flexibility and adaptability but also accelerates the processes of knowledge acquisition and assimilation. When a software development team encounters shifts in requirements, individuals with expertise relevant to the core issues assume leadership roles, while other team members contribute knowledge to foster cross-functional collaboration. This dynamic influences the circulation and absorption of knowledge, thereby enhancing the speed and effectiveness of addressing demand-related challenges. For instance, when new requirements emerge, those with specialized knowledge in that area take the lead, swiftly sharing their insights to guarantee that the new requirements are thoroughly understood and translated into actionable development strategies. This open exchange of information aids in knowledge absorption and mitigates the impact of changes on development timelines. Therefore, this study posits that shared leadership positively influences absorptive capacity and formulates the following hypothesis:

Proposition 1: Shared leadership is positively associated with absorptive capacity

- **The Influence of Absorptive Capacity on Software Development Agility**

Absorptive capacity theory highlights the strong correlation between a firm's dynamic capacity of learning and its pre-existing knowledge base, internal learning mechanisms, and innovation potential (Samaeemofrad and van den Herik, 2020). In the realm of software development agility, it pertains to a software team's capability to adapt swiftly and effectively to evolving user demands (Lee and Xia, 2010). Within a software development context, teams are required to navigate various changes, cultivate a structure

that supports frequent and sustainable operations, and adapt to shifting requirements. Teams exhibiting robust absorptive capacity can swiftly learn new technologies, methodologies, and tools (Sun and Anderson, 2010), and apply these insights to manage fluctuating requirements.

When a software development team possesses strong external knowledge search capabilities and its members actively engage in activities such as participating in technical communities, attending seminars, and enrolling in online courses, the team's overall knowledge acquisition capacity is significantly enhanced. This proactive approach shortens the time required for knowledge gathering and requirements analysis, enabling the team to more rapidly understand evolving changes, identify appropriate solutions (Sun and Anderson, 2010), reduce decision-making delays, and improve the efficiency of responding to new demands. Effective knowledge absorption plays a critical role in reducing internal information asymmetry, thereby facilitating faster collaboration among team members during the development process. This, in turn, minimizes errors and rework while accelerating the use of absorbed knowledge to address emerging challenges. Absorptive capacity equips teams with the ability to master specific external knowledge and modify existing practices accordingly (Lee et al., 2018) Teams characterized by high absorptive capacity can transform external knowledge as well as insights gained from past problems and solutions into valuable internal knowledge assets. Furthermore, following the absorption of new knowledge, the software team can integrate updated skills and technical architecture. The enhancement of absorptive capacity can also foster innovative behavior, allowing for the transformation of previously acquired knowledge into novel insights and solutions (Pangarso et al., 2020). This creates opportunities for knowledge reuse across a broader spectrum, stimulating new innovations, generating fresh solutions, and improving overall response efficiency. Consequently, this study posits that absorptive capacity positively influences the response efficiency of software teams and proposes the following hypothesis:

Proposition 2: Absorptive capacity is positively associated with software team response efficiency

Software development team with strong absorptive capacity can proactively and effectively acquire knowledge from a wide range of external sources, including customers, markets, competitors, and academic research. This broad engagement exposes the team to diverse solutions and emerging technological trends, facilitates information exchange, and enhances mutual understanding among team members. Once new knowledge is acquired, the team assimilates it, enabling a deeper comprehension of customer needs and a more holistic grasp of the problems at hand. This process, in turn, supports the development of superior solutions and products that align more closely with customer expectations. When software teams are confronted with extensive and diverse requirement changes, their ability to flexibly transform knowledge becomes critical. Such flexibility allows teams to adapt solutions to varying circumstances, innovate by drawing on lessons learned from historical case experiences (Pangarso et al., 2020), and generate multiple solution alternatives. Ultimately, the successful utilization

and application of this knowledge enable teams to address a wide range of needs, thereby enhancing the extensiveness of their response capabilities. Absorptive capacity plays a crucial role in facilitating knowledge sharing among software teams, enabling a more comprehensive understanding of challenges and a thorough analysis of requirements when confronted with diverse demand changes. This, in turn, leads to the proposal of a wider array of solutions (Cohen and Levinthal, 1990), thereby broadening the scope of responses. In light of this, the study posits that absorptive capacity positively influences the extensiveness of responses from software teams and presents the following hypothesis:

Proposition 3: Absorptive capacity is positively associated with software team response extensiveness

Based on the above development, a model is proposed and is shown in Fig. 1 below.

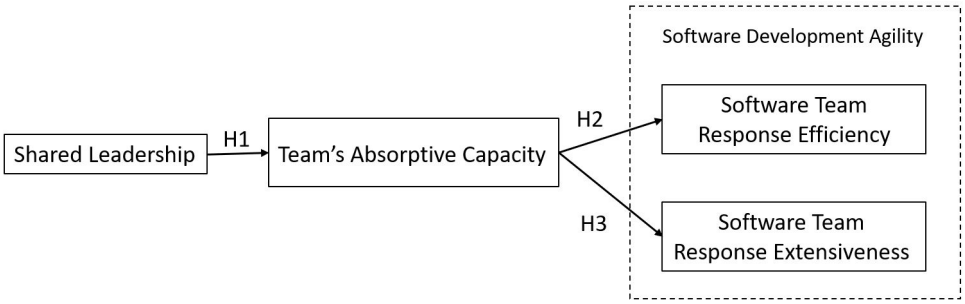

Figure 1: The proposed conceptual model.

CONCLUSION & WHAT'S NEXT

This paper is an interdisciplinary study that innovatively utilizes absorptive capacity to observe how team dynamics and leadership in the knowledge-intensive requirements changes context facilitates software development agility. As a preliminary result, a conceptual model is proposed to theorize the effects of shared leadership and team absorptive capacity on the performance of software development agility. The model is conceptual in nature, thus in the next study, we would like to conduct an empirical investigation to realize the theoretical model proposed in this paper to receive a broader and generalized aspect of the results from samples to the population of interest. Specifically, we will take the survey method, in which empirical data are collected from software development teams that have exercised and possess experiences of process tailoring. Due to the scope of the survey that is at the team level, single-source bias may occur and thus matched pairs (i.e., different informants for different constructs) are suggested in the participant design. As for the statistical techniques for analysis, structural equation modelling (SEM) may be appropriate for testing theoretical relationships to obtain more quality results.

As for other future research, we suggest that future research extension can take the following directions. Firstly, the proposed model and propositions

can be further studied using case-based approach. A case research method is useful when the empirical study expects a more in-depth exploration that is from the operational aspect. In conducting this method, we suggest that data collection should consider the issue of triangulation. That is, the empirical evidences can be collected from at least three different sources within the studied case. For example, data can be acquired from (1) archives, such as project documentations, tailoring technical reports and customer records, (2) onsite observations on tailoring meetings, work activities and team conversations to obtain elaborate field notes, and (3) interviews to obtain data for analysis. As for the validity of data when conducting multiple cases, it is suggested to perform a cross analysis to compare the practices across the teams to ensure the inter-consistency of the results.

REFERENCES

Ali, I., Musawir, A. U., & Ali, M. (2018). Impact of knowledge sharing and absorptive capacity on project performance: The moderating role of social processes. *Journal of knowledge management*, *22*(2), 453–477.

Baham, C., & Hirschheim, R. (2022). Issues, challenges, and a proposed theoretical core of agile software development research. *Information Systems Journal*, *32*(1), 103–129.

Bjørnson, F. O., & Dingsøyr, T. (2008). Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and software technology*, *50*(11), 1055–1068.

Chen, C. Y. & Lee J. C. (2025). The effects of learning from failure and team cognitive factors on software process tailoring. *ACM Transactions on Software Engineering and Methodology*. doi: 10.1145/3747179.

Chow, T., & Cao, D.-B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, *81*(6), 961–971.

Cohen, W. M., & Levinthal, D. A. (1990). Absorptive capacity: A new perspective on learning and innovation. *Administrative Science Quarterly*, *35*(1), 128–152.

Cullen-Lester, K. L., & Yammarino, F. J. (2016). Collective and network approaches to leadership. *The Leadership Quarterly*, 27(2), 173–180.

Eisenhardt, K. M., & Martin, J. A. (2000). Dynamic capabilities: What are they? *Strategic Management Journal*, *21*(10–11), 1105–1121.

García-Morales, V. J., Jiménez-Barrionuevo, M. M., & Gutiérrez-Gutiérrez, L. (2012). Transformational leadership influence on organizational performance through organizational learning and innovation. *Journal of Business Research*, *65*(7), 1040–1050.

Gronn, P. (2002). Distributed leadership as a unit of analysis. *The Leadership Quarterly*, *13*(4), 423–451.

Hoch, J. E., & Dulebohn, J. H. (2017). Team personality composition, emergent leadership and shared leadership in virtual teams: A theoretical framework. *Human Resource Management Review*, *27*(4), 678–693.

Kimura, T. (2024). Virtual teams: a smart literature review of four decades of research. *Human Behavior and Emerging Technologies*, 2024(1), 8373370.

Langfred, C. W. (2000). The paradox of self-management: Individual and group autonomy in work groups. Journal of Organizational Behavior, 21(5), 563–585.

Laplante, P. A., & Kassab, M. (2022). *Requirements Engineering for Software and Systems*. Auerbach Publications.

Lee, G., & Xia, W. (2010). Toward agile: An integrated analysis of quantitative and qualitative field data on software development agility. *MIS quarterly*, *34*(1), 87–114.

Lee, J. C., Chou, I. C., & Chen, C. Y. (2021). The effect of process tailoring on software project performance: The role of team absorptive capacity and its knowledge-based enablers. *Information Systems Journal*, *31*(1), 120–147.

Lee, J. C., Hsu, W. C., & Chen, C. Y. (2018). Impact of absorptive capability on software process improvement and firm performance. *Information Technology and Management*, *19*, 21–35.

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.

Pangarso, A., Astuti, E. S., Raharjo, K., & Afrianty, T. W. (2020). The impact of absorptive capacity and innovation ambidexterity on sustainable competitive advantage: The case of Indonesian higher education. *Entrepreneurship and Sustainability Issues*, *7*(3), 2436.

Rathor, S., Xia, W., & Batra, D. (2024). Achieving software development agility: Different roles of team, methodological and process factors. *Information Technology & People*, *37*(2), 835–873.

Samaeemofrad, N., & van den Herik, J. (2020). A moderating role of absorptive capacity within incubation support. 2020 IEEE international conference on engineering, technology and innovation (ICE/ITMC).

Sancho-Zamora, R., Hernández-Perlines, F., Peña-García, I., & Gutiérrez-Broncano, S. (2022). The impact of absorptive capacity on innovation: The mediating role of organizational learning. *International Journal of Environmental Research and Public Health*, *19*(2), 842.

Spagnoletti, P., Kazemargi, N., & Prencipe, A. (2021). Agile practices and organizational agility in software ecosystems. *IEEE Transactions on Engineering Management*, *69*(6), 3604–3617.

Strode, D., Dingsøyr, T., & Lindsjorn, Y. (2022). A teamwork effectiveness model for agile software development. *Empirical Software Engineering*, 27(2), 27–56.

Sun, P. Y., & Anderson, M. H. (2010). An examination of the relationship between absorptive capacity and organizational learning, and a proposed integration. *International Journal of Management Reviews*, *12*(2), 130–150.

Teece, D. J (2012). Dynamic Capabilities: Routines Versus Entrepreneurial Action. *Journal of Management Studies*, 49(8), 1395–1401.

Wang, D., Waldman, D. A., & Zhang, Z. (2014). A meta-analysis of shared leadership and team effectiveness. *Journal of Applied Psychology*, *99*(2), 181.

Xu, N., Ghahremani, H., Lemoine, G. J., & Tesluk, P. E. (2022). Emergence of shared leadership networks in teams: An adaptive process perspective. *The Leadership Quarterly*, 33(6), 101588.

Zollo, M., & Winter, S. G. (2002). Deliberate learning and the evolution of dynamic capabilities. *Organization Science*, *13*(3), 339–351.