International Research Workshop on IT Project Management 2019

International Research Workshop on IT Project Management (IRWITPM)

12-14-2019

Project Management Tools as Boundary Objects in Agile Software Development

Yuanyanhang Shen

Xiaodan Yu

Project Management Tools as Boundary Objects in Agile Software Development

Yuanyanhang Shen
University of International Business and Economics
yanhangshen@163.com

Xiaodan Yu
Corresponding author
University of International Business and Economics
xyu@uibe.edu.cn

ABSTRACT

In agile software development (ASD) teams, it is essential to overcome knowledge boundaries to prevent product delays. The theory of boundary objects suggests that using the objects can help bridging knowledge boundaries within ASD teams in collaborations. Although prior research has reported that the use of boundary objects within traditional software development (TSD) teams is helpful, this topic in agile background still needs more exploration. Additionally, findings on the effects of boundary objects in bridging knowledge gaps are mixed. In this in-progress study, we conceptually explored the role of project management tools as boundary objects in ASD teams. Empirical study was conducted by using eight student teams, each consisting of four to five team members, which were asked to deliver a software using project management tools. Preliminary data analysis showed that PMTs indeed have positive influence in agile context.

Keywords

Agile software development, boundary objects, knowledge sharing.

INTRODUCTION

According to the agile manifesto pronounced in 2001[1], software developers are encouraged to deliver working software to customers regularly in order to accommodate requirement changes and deliver best products (Dingsøyr, Nerur, Balijepally and Moe, 2012). Effective communication and knowledge sharing between technicians and end users are therefore essential for the success of agile projects (Chan and Thong, 2009). Nevertheless, these projects often suffer from the huge discrepancy of knowledge boundaries among actors (Tiwana and Mclean, 2005). Ineffective bridging these knowledge gaps can result in project delays and other negative project outcomes (Levina and Vaast, 2008).

The concept of boundary objects is a promising theoretical lens in facilitating collaboration across knowledge boundaries within ASD teams. Boundary objects are defined as shared or sharable objects across different problem solving settings (Carlile, 2002; Star and Griesemer, 1989). Researchers have demonstrated that boundary objects, such as PMTs, have both bright and dark sides during the tasks. (Barrett and Oborn, 2010). According to Carlile (2002), boundary objects are not "magic bullets" because their characteristics will change due to different problems and people. Therefore, it is necessary to re-examine the use of boundary objects in agile settings. ASD is totally different from TSD due to the

[1] http://agilemanifesto.org/

attribute of agile that software development processes are divided into short iterations in which new requirements probably emerge (Ramesh, Cao and Baskerville, 2010).

Though some studies have been done in exploring boundary objects within TSD, unfortunately, study falls short in shedding further light on the deep understanding of boundary objects and ASD. One exceptional study was conducted by Huber, Winkler, Dibbern and Brown (2019) . They found that a fit between the type of knowledge boundaries, software prototype characteristics, and the use practices can lead to effective bridging of knowledge boundaries in ASD teams. Inspired by their work, it is our study interest to examine the mechanism of using project management tools as effective boundary objects in ASD. The literature on project management tools (such as Microsoft Project and Teambition), suggests numerous roles like visualization of the project status, executing control on the project teams, etc. (Hebert and Deckro, 2011). But our knowledge of how PMT facilitates knowledge sharing in the ASD setting is still lacking.

We therefore seek to address this question, "How does the use of PMTs as boundary objects help to bridge knowledge boundaries within ASD teams?"

In this RIP paper, we theoretically explore the use of PMTs as boundary objects within ASD teams. In the first section, we examine previous literature. Next, a brief description of our methods is made. After that, we present the preliminary analysis of results. Finally, we describe the future research plan.

CONCEPTUAL FOUNDATION

Agile Software Development (ASD)

The agile methods are different from traditional development related to the assumptions of requirement engineering processes (Sillitti, Ceschi, Russo and Succi, 2005; Turk, France and Rumpe, 2005). For example, agile methods assume that requirements evolve over time, while traditional development consider that customer can specify all their needs in the initial phase. Additionally, ASD counts on interactions between customers and developers to specify requirements which can emerge during each iteration, while traditional development relies on interactions among several stakeholders (Turk et al., 2005).

Studies, however, have consistently demonstrated that it is challenging for software development teams to share knowledge and collaborate effectively (Barrett and Oborn, 2010; Chan and Thong, 2009; Tiwana and Mclean, 2005). For instance, Tiwana and Mclean (2005) indicated that knowledge sharing is problematic because of diverse "domain specific knowledge". Besides, Rosenkranz, Vranešić and Holten (2014) demonstrated that the "differing perspectives" or "different experience frames" between users and technical experts can lead to failure in requirement elicitation phases. In consequence, effective knowledge sharing and collaboration across boundaries is arduous (Levina and Vaast, 2008).

Compared to traditional software development, ASD faces even more challenges. According to Chan and Thong (2009), agile principles place a greater emphasis on teams' collaboration and the ability to respond to rapid changing requirements compared to traditional software development. ASD teams usually face more difficulties in time scheduling (Ramesh et al., 2010). These fast-changing requirements need instant response by the ASD members, and they have to communicate with users under time pressure. This calls for research to explore the issue and help ASD teams improve collaboration performance especially on the project management aspects.

Shared Mental Models (SMMs)

The definition of shared mental model (SMM) was team members' shared, organized understanding and mental representation of knowledge about key elements of the team's relevant environment (Klimoski and Mohammed, 1994). Empirical research has tended to propose two major SMMs content domains: teamwork mental model (TMMM) and taskwork mental model (TKMM). TMMM focuses on interpersonal interaction, while TKMM includes work goals and performance requirements (Cannon-Bowers et al. 1993).

According to the theory of SMMs, a team level mental model can be developed after each member has established his or her mental model through collaboration (Badke-Schaub, Neumann, Lauche and Mohammed, 2007). An effective communication mechanism therefore is a key element in building SMMs within a team(Cannon-Bowers, Salas and Converse, 1993). Previous study has explored TKMMs and TMMMs in traditional software development teams and concluded that it could improve the quality of the final product (Hsu, Chang, Klein and Jiang, 2011; Rai, Maruping and Venkatesh, 2009; Yang, Kang and Mason, 2017).

However, ASD projects often suffer from large asymmetries of knowledge among actors (Tiwana and Mclean, 2005), raising challenges to the effective bridging of knowledge boundaries and communication. As a result, ASD teams can have much more difficulties in building SMMs. Prior research suggests that ASD teams can bridge their knowledge boundaries and establishing SMMs through effectively choose various ASD practices (Yu and Stacie, 2014). However, our knowledge of how project management tools support the development of SMMs is limited.

Knowledge Boundary and Boundary Objects

Knowledge sharing across boundaries is challenging and significant within ASD teams. Carlile (2002) identified three types of knowledge boundaries: syntactic, semantic, and pragmatic. To deal with knowledge boundaries, one can employ boundary objects (Star, 2010). Boundary objects are defined as shared or sharable objects across different problem solving settings (Carlile, 2002; Star and Griesemer, 1989).

Studies have found mixed effect of boundary objects (Barrett and Oborn, 2010; Sapsed and Salter, 2004). This can be attributed to the fact that boundary objects only influence knowledge sharing across boundaries if they are accommodated to work practices (Huber et al., 2019; Levina and Vaast, 2005). If the boundary objects are appropriately utilized, they can help to transfer and translate knowledge (Carlile, 2004), to accommodate interests (Levina and Vaast, 2005), and to overcome cultural differences (Barrett and Oborn, 2010).

To take full advantage of boundary objects, Doolin and McLeod (2012) indicated that the context where boundary objects are embedded in is important, because the practices of the objects are highly sensitive to context. As ASD is greatly distinct from TSD, studying the use of boundary objects in agile settings promises novel insights on the function of the objects for bridging knowledge boundaries.

The theory of boundary objects permit a theoretical explanation of the underlying dimension of how ASD members develop their SMMs. Carlile (2004) indicated that boundary objects could transfer, translate and transform knowledge at syntactic, semantic and pragmatic boundary. Transferring knowledge at the syntactic boundary enable teams to build a shared lexicon among team members. For instance, in ASD teams, consumers transfer their requirements to the developers at this boundary using the "language" they developed. This process helps teams possess information, manage knowledge, and have a rough stretch of the task or the goals, enabling the emergence of TKMMs. We argue that:

Hypothesis 1: ASD teams' syntactic boundaries negatively influence the development of TKMMs in ASD teams.

At semantic boundary, team members' translations or interpretations diverge, raising barriers to the create both TMMMs and TKMMs. This is because even when team members share the same terminology, they tend to translate the meaning of the knowledge differently. For example, when the bridging of semantic boundary fails, developers can misunderstand users' requirements and make an unappropriated software development plan, resulting in obstacles during the establishment of TKMMs. Besides, at this boundary, users can also have difficulties in truly understanding the roles, skills or responsibilities of various technical experts which probably lead to ineffective development of TMMMs.

Hypothesis 2a: ASD teams' semantic boundary negatively influence the development of TKMMs in ASD teams.

Hypothesis 2b: ASD teams' semantic boundary negatively influence the development of TMMMs in ASD teams.

At pragmatic boundary, teams can employ boundary objects to build shared interests and transform knowledge. If ASD teams fail to bridge pragmatic boundary, they will be unable to negotiate, make trade-off, and reach a consensus among teams to adequately share and assess knowledge (Carlile, 2004). In other words, members at this boundary will have conflicting opinions on others' roles and responsibilities, which are involved in TMMMs. In ASD teams, users are more interested in the product features and cost, while developers have stronger interest on technical elegance. Thus, this conflict interest raises barriers for the ASD teams to converge those preferences. If the pragmatic boundaries are not bridged, it is less possible for the teams to establish TMMMs and find a consensus leading to project success. Consequently, we argue that:

Hypothesis 3: ASD teams' pragmatic boundaries will negatively influence the development of TMMMs in ASD teams.

Project Management Tools (PMTs) as Boundary Objects in ASD

Many studies have explored PMTs in traditional software development teams. Inconsistent findings exist in the strand of research. Specifically, Yakura (2002) described how could timelines—a specific project management tool — function effectively as a boundary object to coordinate diverse temporal arrangements. Sapsed and Salter (2004) indicated that as boundary objects, PMTs' effect is limited for geographically dispersed teams when the authority and control were ambiguous or missing. According to Barrett and Oborn (2010), the use of the same PMTs could both help and limit the processes of knowledge sharing across boundaries in the same environment over time. It can be challenging to employ these tools even during traditional software development. This mixed findings can be attributed to the fact that boundary objects only influence knowledge sharing across boundaries if they are accommodated to work practices (Huber et al., 2019; Levina and Vaast, 2005).

PMTs as boundary objects can be most useful in reducing different knowledge gap among team members. For example, in a database, terms and information regarding task and team can be searched. Accordingly, a database can be most useful in bridging the syntactic boundary (Carlile, 2004). Timelines can graphically represent scheduling and task assignment, reducing the uncertainty and ambiguity in the translation of scheduling and enabling the development of TKMMs among team members. If timelines are appropriately used to ease time pressure, team members may pay more attention to the inter-team communication rather than mere task accomplishment. Conflict of interests would probably be identified earlier, which help bridge the pragmatic boundary (Barrett and Oborn, 2010). If PMTs are not accommodated to work settings, we name that this is a "misuse" of PMTs. Therefore, we propose the following hypotheses.

Hypothesis 4a: The more severe the misuse of PMTs is, the less bridged the syntactic boundaries will be.

Hypothesis 4b: The more severe the misuse of PMTs is, the less bridged the semantic boundaries will be.

Hypothesis 4c: The more severe the misuse of PMTs is, the less bridged the pragmatic boundaries will be.

RESEARCH DESIGN

Building on the conceptual foundation, we aim to build theory on how the use of PMTs as boundary objects contribute to the bridging of knowledge boundaries in agile software development. To achieve these goals, we have conducted an experiment and finished our data collection.

Procedure and Data Collection

First, we recruited thirty-three study participants from a university in China. All of them had taken the prerequisite courses and were able to participate in ASD projects. The student sample is appropriate because in practices, not all stakeholders have the experience of ASD projects.

We assigned them randomly into eight groups. Each group had four to five students. The group task was to develop an information system, which was complex to ensure enough collaboration. Students were asked to follow a rough agile instruction provided by one of the authors of this paper. Face-to-face communication, WeChat (a popular social communication tool in China), and PMTs were employed by the teams to communicate and discuss. Before the implementation of the ASD task, questionnaires were administrated to students. Interviews were also conducted at this time. Finally, when all teams finished their tasks, instructors evaluated team performance objectively. Questionnaire and interview questions were mainly about students' perceptions on team knowledge boundaries and students' reflections on how they used the project management tools and how the PMTs help their achieving various team related activities.

Preliminary Analysis and Findings

The preliminary analysis primarily developed initial open codes from the open-end questions in the questionnaires during ASD. Eight teams were coded from 1 to 8 respectively. Through interviews and observations, we found that groups presented different levels of satisfaction with PMTs. Participants who reported positive effect of PMTs mainly expressed the reinforcement of time management, task assignment, communication and documentation. Some students, however, experienced low-efficiency collaboration.

For example, compared to Team 6, Team 2 reported a higher level of average satisfaction. Members from team 2 shared similar opinions relating to the time management and task assignment, while students from team 6 held various understandings on the issue. In other words, knowledge boundaries failed to be bridged in team 6. Based on the answers from team 6, this can be attributed to the fact that they were unable to recognize others' task status, indicating inappropriate use of PMTs as boundary objects.

FUTURE RESEARCH PLAN AND EXPECTED CONTRIBUTIONS

Various boundary objects have been examined in traditional software development. Research on the use of PMTs as boundary objects in ASD, however, is still unexplored. This RIP paper aims to bridge this gap by providing a theoretical exploration. A preliminary analysis indicated that problems indeed existed in this environment.

For future research plan, NVivo will be utilized to track and code qualitative research to empirically address these problems. We will systematically analyze the data collected. Additionally, we will perform another experiment to examine different PMTs in Scrum contexts to further our understanding the use of PMTs in agile.

Through this qualitative research, rich insight on the role of the PMTs as boundary objects in ASD teams is expected. Through the lens of SMMs theory, we can uncover the black box of the use of PMTs as boundary objects in ASD teams, theoretically contributing to the extant study. Additionally, this research will provide practical guidelines on how to use PMTs to manage ASD project. As prior research has suggested that ASD teams may meet more barriers in using PMT as boundary objects. This is because ASD teams work in short iterations or "sprints", in which software requirements will be revised (Ramesh et al., 2010). Thus, it is even harder for teams to maintain the PMTs and take full advantage of these objects. Findings from our research has the potential to expand our understanding of how to utilize PMTs as boundary objects to bridge knowledge boundaries in ASD.

REFERENCES

Badke-Schaub, P., Neumann, A., Lauche, K., and Mohammed, S. (2007) Mental models in design teams: A valid approach to performance in design collaboration?, *CoDesign*, 3, 1, 5-20.

Barrett, M., and Oborn, E. (2010) Boundary object use in cross-cultural software development teams, *Human Relations*, 63, 8, 1199-1221.

Cannon-Bowers, J. A., Salas, E., and Converse, S. (1993) Shared mental models in expert team decision making, in: J. N. J. Castellan (Ed.), Individual and group decision making: Current issues, Lawrence Erlbaum Associates, Hillsdale, NJ, US, 221-246.

Carlile, P. R. (2002) A pragmatic view of knowledge and boundaries: Boundary objects in new product development, *Organization Science*, 13, 4, 442-455.

Carlile, P. R. (2004) Transferring, translating, and transforming: An integrative framework for managing knowledge across boundaries, *Organization Science*, 15, 5, 555-568.

Chan, F. K. Y., and Thong, J. Y. L. (2009) Acceptance of agile methodologies: A critical review and conceptual framework, *Decision Support Systems*, 46, 4, 803-814.

Dingsøyr, T., Nerur, S., Balijepally, V., and Moe, N. B. (2012) A decade of agile methodologies: Towards explaining agile software development, *Journal of Systems and Software*, 85, 6, 1213-1221.

Doolin, B., and McLeod, L. (2012) Sociomateriality and boundary objects in information systems development, *European Journal of Information Systems*, 21, 5, 570-586.

Hebert, J. E., and Deckro, R. F. (2011) Combining contemporary and traditional project management tools to resolve a project scheduling problem, *Computers & Operations Research*, 38, 1, 21-32.

Hsu, J. S. C., Chang, J. Y. T., Klein, G., and Jiang, J. J. (2011) Exploring the impact of team mental models on information utilization and project performance in system development, *International Journal of Project Management*, 29, 1, 1-12.

Huber, T. L., Winkler, M. A. E., Dibbern, J., and Brown, C. V. (2019) The use of prototypes to bridge knowledge boundaries in agile software development, *Information Systems Journal*.

Klimoski, R., and Mohammed, S. (1994) Team mental model: Construct or metaphor?, *Journal of Management*, 20, 403-437.

Levina, N., and Vaast, E. (2005) The emergence of boundary spanning competence in practice: Implications for implementation and use of information systems, *MIS Quarterly*, 29, 2, 335–363.

Levina, N., and Vaast, E. (2008) Innovating or doing as told? Status differences and overlapping boundaries in offshore collaboration, *MIS Quarterly*, 32, 2, 307-332.

Rai, A., Maruping, L., and Venkatesh, V. (2009) Offshore information systems project success: The role of social embeddedness andcultural characteristics, *MIS Quarterly*, 33, 3, 617-641.

Ramesh, B., Cao, L., and Baskerville, R. (2010) Agile requirements engineering practices and challenges: An empirical study, *Information Systems Journal*, 20, 5, 449-480.

Rosenkranz, C., Vranešić, H., and Holten, R. (2014) Boundary interactions and motors of change in requirements elicitation: A dynamic perspective on knowledge sharing, *Journal of the Association for Information Systems*, 15, 6, 306-345.

Sapsed, J., and Salter, A. (2004) Postcards from the edge: Local communities, global programs and boundary objects, *Organization Studies*, 25, 9, 1515-1534.

Sillitti, A., Ceschi, M., Russo, B., and Succi, G. (2005) Managing uncertainty in requirements: A survey in documentation-driven and agile companies, *11th IEEE International Software Metrics Symposium (METRICS'05),* September 19-22, 10 -17.

Star, S. L. (2010) This is not a boundary object: Reflections on the origin of a concept, *Science, Technology & Human Values*, 35, 5, 601–617.

Star, S. L., and Griesemer, J. R. (1989) Institutional ecology, 'translations' and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907-39, *Social Studies of Science*, 19, 3, 387-420.

Tiwana, A., and Mclean, E. R. (2005) Expertise integration and creativity in information systems development, *Journal of Management Information Systems*, 22, 1, 13-43.

Turk, D., France, R., and Rumpe, B. (2005) Assumptions underlying agile software-development processes, *Journal of Database Management*, 16, 62-87.

Yakura, E. K. (2002) Charting time: Timelines as temporal boundary objects, *Academy of Management Journal*, 45, 5, 956-970.

Yang, H.-D., Kang, H.-R., and Mason, R. M. (2017) An exploratory study on meta skills in software development teams: Antecedent cooperation skills and personality for shared mental models, *European Journal of Information Systems*, 17, 1, 47-61.

Yu, X., and Stacie, P. (2014) Understanding agile software development practices using shared mental models theory, *Information and Software Technology*, 56, 8, 911-921.