Technology Frame Disruption During Short-Term Agile ISD Projects

Aaron Read
Arizona State University
Aaron.Read@asu.edu

Joseph C. Clark
Arizona State University
Joseph.W.Clark@asu.edu

Abstract

Effective collaboration between clients and development teams is vitally important to all Agile ISD methods, enabling the key benefit of Agile approaches which is the ability to react quickly to changing requirements. The process by which this collaboration develops is particularly consequential for short-term Agile projects where we cannot assume that there will be sufficient time for initial "kinks" to be worked out. We develop a process model, based on extended concepts of technology frames and technology frame disruption, to explore the extent to which client framing of software projects is resistant to change in a highly collaborative environment. We report on survey and interview data from fourteen Agile ISD project clients, and discuss continuing research. We find that novice clients' frames focus on business-facing technology characteristics, but do not focus on project considerations, most notably the need to understand the capabilities of the team, and the technical constraints of the technology.

1. Introduction

1.1. Collaboration building in Agile ISD

Agile information systems development (ISD) methods are effective means of responding to requirements change, both the changes that come from outside of the ISD process – changing market demand or client needs – and those that emerge during the work, such as design and engineering changes which occur as a result of an increased understanding of the problem the requirements are trying to solve. Agility in ISD, which combines aspects of flexibility and leanness [1], has been shown to positively affect project outcomes [2], and to improve team performance by enabling teams to act as autonomous, collaborative agents [3].

Effective collaboration between clients and developers is vitally important to all Agile ISD methods, which involve clients in planning, prioritizing requirements, and giving feedback on features as they are developed [4], and lack of such collaboration reduces the team's ability to react to new knowledge. When requirements are difficult to identify, frequent interaction enables clients and developers to iteratively gain understanding of the requirements and the solution that will satisfy them.

It often takes time for new collaborations to develop. At first, clients may find it difficult to understand and articulate their own needs in ways that make sense to developers, and developers may not know how to share their technical expertise in terms of the affordances or constraints the tools can create for the business. In the long run, these problems may be mitigated through a process of mutual learning.

The nature of this learning process is, however, particularly consequential for clients and developers working together on short-term Agile projects, or projects that may be at risk of cancellation based on short-run results, where we cannot assume that there will be a "long run" in which to get client-developer collaboration working. The present study seeks to expand our knowledge of how new collaborations between clients and development teams form in the short run of a new Agile ISD project. Such an understanding is crucially important for us to be able to anticipate the challenges faced by practitioners, and perhaps help them to bootstrap effective collaboration in fewer iterations.

1.2. Technology frames as a research lens

The client's and developer's knowledge of requirements is largely tacit in nature – expertise which arises from repeated use of a technology becoming an automated response. The concept of *technology frames* encapsulates this expertise, along with other assumptions as a lens with which the individual utilizes this technology. As the individual or group considers the technology, the frame will make some information salient while masking other information. Individual technology frames may be shaped by affordances—perceived uses associated

1530-1605/15 $31.00 © 2015 IEEE
DOI 10.1109/HICSS.2015.605

with the technology. On the other hand, when an individual's goals are not adequately served by the technology, the constraints of the technology become salient. The individual may misuse or appropriate the technology to suit their needs, or seek to modify the technology.

These technology frames are resistant to change, but can be disrupted as "breakdowns" occur. These breakdowns may occur when the technology frame of an individual or group is incompatible with another stakeholder's technology frame. While these disruptions can have a negative effect on a software development project [5], they may also be necessary for the formation of shared understanding and compromise in a systems development effort [6].

We use a concept of technology frames which includes not only the framing of the technology itself, but also the framing of the project process, as this may be useful in understanding the development of shared understanding in Agile teams as clients collaborate with developers. We call this a technology project frame.

Developers and clients often experience a communication gap, where enabling clients to transfer knowledge to developers is difficult to achieve. Previous research in requirements elicitation has focused on identifying cognitive mechanisms which both inhibit and enable knowledge transfer from a client to an analyst [7-11]. While this research increases our understanding of cognitive aspects, we are left uninformed about the *content* of the knowledge frames of systems development participants, especially which types content are likely to hinder the performance of software development teams. With such knowledge, common assumptions and expectations associated with these frames could be identified and made salient to improve development team performance.

The current research seeks to identify the common frames held by clients in the beginning stages of development of new software product. Specifically, we seek to identify common content in these frames as well as the patterns of frame disruption and learning which occur as the project teams cycle through several initial development iteration. Such understanding could lead to the improvement of collaboration and client learning which are crucial to the performance of software development teams in an Agile setting.

To investigate the content of the technology project frames and subsequent learning and adaptation, we conducted a field study of the capstone projects of 27 student teams working with real clients. We present the theoretical background for our hypotheses, followed by an explanation of our study methodology and analysis. We conclude with a discussion of potential findings and directions for future research.

2. Background

2.1. Technology frames

Technology frames describe how individuals and organizational groups [12-15] interpret technology in its development and use [5,16]. They include the beliefs, assumptions, and values which are used to frame problem solving. They can both mask inconsistent information and call attention to or invent information which conforms with existing knowledge structures [17]. Frame incongruence between social groups such as business clients and developers, can be used to explain software project failure [5,16]. Ultimately, when technology frames are unable to converge, or stabilize [6], the project suffers inefficiencies and may fail when disagreements about the software lead to constant requirements changes.

While developing an innovative product, an individual or group may be unaware of or "blind" to the fact that others may be using different technology frames as they contemplate the software product [18]. Developers can simplify or "black box" the socio-cognitive framing of other individuals [19], leading them to avoid the pursuit of what is really valuable and instead seek to be faithful to a method. This is exacerbated by the fact that developers can be strained in Agile settings to wear many hats [20], and may not develop the communication skills necessary.

Clients, on the other hand, often assume that the developer is aware of their needs, and leave potentially important constraints of the requirements undiscussed or poorly discussed [21-23]. Thus, there is a communication feedback loop gap that often exists in software development projects [24].

2.2. Technology project frames

Technology frames have been used in multiple domains—specific groupings of contexts which relate in different ways to the development of technology. We introduce the term *technology project frames* to include not only beliefs, assumptions, and so on about the technology itself [5,18,26], but also about how to proceed with the project. These technology project frames are also stable, but can include shifts as the result of the decisions of powerful individuals, exposure to new technical and business strategies and solutions, and technological artifacts. Frame

adjustments are a necessary part of the process of developing software [27], as they bring new knowledge to the forefront of discussion by reframing the problem to solve. Frame adjustments may be necessary to integrate business and technical knowledge [28]. It is following the learning that happens in these divergent stages that shared understanding can be reached [29].

When parties holding differing frames are both aware of the competing frame and willing to adjust, a more stable frame can emerge. Azad and Faraj [6] describe a technology stabilization process which begins with *frame differentiation*, where frames solidify into competing frames which provide mutual obstacles for one another, *frame adaptation*, where the members of groups holding opposing frames negotiate and develop a "truce frame," and *frame stabilization* where the truce frame evolves to stability, and is perceived as taken for granted.

2.3. Technology frame disruption and learning in Agile software development

Agile software development methods provide frequent opportunities for the client and developer technology frames to interact through continuous communication and the iterative delivery of functionality to the client. Agility enables a software team to immediately incorporate change through emergent control [30]. Emergent control implies that design and specification work does not heavily outpace development, to enable the necessary responsiveness to potential changes. Thus, if there are disparities in the client frame and the developer frame, they will likely be revealed slowly.

In some instances, a client may not perceive the emerging changes, and therefore not comprehend the implications for their framing of the new technology. They will therefore not be able to provide feedback in early stages of the development and later complain that the software does not meet their needs [31]. This lack of perception can originate from several sources, including overhype of the Agile process [4], lack of communication, or an inability to articulate what they want [23]. On the other hand, with increased up-front investment in the design process, the client may have a more positive perception of interaction in the design process [32]. Ultimately, if the client is able to provide feedback earlier in the development of the software, the necessary negotiating and learning may occur for both the developers and the client to arrive at a stable frame for the technology.

2.4. A tentative process model of client and developer learning through frame disruption

Because of the emphasis placed on the collaborative and adaptive nature of Agile software development teams, we seek to determine whether or not technology framing can explain why a developer/client team is unable to adjust quickly in a short frame of time. We use the following theoretical framework for our investigation (also see Figure 1).

When working on new projects in a short time frame, we assume that frame disruption will occur in a manner similar to those discussed in previous literature [5,18,26]. Most notably, previous research suggests that the client will begin with a focus on pre-existing constraints and desired affordances [18], both with respect to the technology and technology projects. Several technology constraints will not be salient to the client at the beginning of the project.

As the development team progresses iteratively to develop the software, the client will learn about new constraints which may or may not be congruent with their current framing of the technology project. Even when aware of the new constraints, the client may not be willing to incorporate them into their framing of the technology project. This willingness may depend on the client's attachments to the affordances of a previous technology, or a previous way of working with a project team.

Even when the client is willing to adjust their frame to incorporate the new constraints, they may or may not perceive that they are able to adjust—because scope is too limited, or they do not perceive that adjustments to the project norms can be made.

1. Client focuses on pre-existing constraints and desired affordances in initial framing.
2. Developers implement the technology frame and present alternative technologies with alternative constraints
3. Client becomes aware of the new constraints and does or does not incorporate them into a new frame.
4. The client perceives new constraints in their ability to implement the newly incorporated constraints within the time frame.
5. The client makes appropriate adjustments to the frame.

Figure 1. Sequence of learning in Agile software development teams

3. Research Context and Design

3.1. An Agile capstone course

The context of this research is the capstone course for the undergraduate computer information systems (CIS) degree program at a major public university in the United States. The course is taken by all senior undergraduates in the program, usually in the spring semester of their final year. There are smaller cohorts in the summer and fall semesters. While the content of lectures pertains to project management methodologies, the true focus of the course is the capstone projects in which student teams work with outside organizations to carry out real-world information systems development projects.

The data collected for this study came from the spring and summer 2014 semesters. In the spring 2014 semester (January to May, 2014), 102 students were enrolled in the course, forming 22 teams working on projects for 21 organizations. In the summer 2014 semester, 26 students formed 5 teams for 4 organizations. Some descriptive data about the teams and projects follows:

Table 1. Team Size

# of students	# of teams
4	12
5	12
6	2
7	1
Total # of teams:	27
Total # of students:	128 (at end of semester)

Table 2. Project Types

project type	# of teams
new website	13
website features	4
new mobile app	3
spreadsheet macro programming	1
data analytics programming	1
Total, software development	22
requirements analysis	3
digital marketing strategy	2
Total, "consulting"	5

Table 3. Client Types

client type	# of clients
entrepreneurial venture	18
established business	4 (6 projects)
nonprofit organization	3
Total # of clients:	25 (two had 2 projects)

As Table 1-3 shows the typical project was 4-5 students working for an entrepreneurial venture[1] on a new website or new features for an existing website, but there was some diversity in the projects. 22 projects could be considered to be software development projects, 3 were requirements analyses for future software development, and the remaining 2 teams worked on digital marketing strategies for their clients.

The spring cohort students were spread out over three sections of the course, and the summer cohort were together in a single section. All sections had the same instructor, one of this paper's authors. Beginning in the spring 2014 semester, the instructor decided to structure the course in accord with the Scrum framework for agile software development, with four "sprints" (iterations) of 2½ weeks each (beginning on Tuesdays, ending two weeks from the following Thursdays). For the summer cohort, there were six weekly sprints. Each iteration began with a planning meeting in which the prioritized requirements backlog was considered and the highest-priority tasks selected for the upcoming iteration, and ended with two meetings: a sprint review meeting in which students demonstrated the current status of their deliverables to their classmates, and a retrospective meeting in which they identified action items for improving their own working process in subsequent iterations. Each team in the spring cohort chose a student ScrumMaster, this requirement was dropped in the summer; in most cases, the client served as Product Owner.[2] Before spring 2014, the course had been structured along a more traditional project timeline, with milestone reports assigned, and a single final deadline for project completion. The

[1] In the "entrepreneurial venture" category, our clients were mostly new start-ups less than one year old and with few or no employees. A few of these clients had a longer history but we did not consider them large or stable enough to be called "established" businesses. Some subjective judgment was used in compiling the table.

[2] ScrumMaster and Product Owner are distinct roles defined in the Scrum framework. In rough outlines, the ScrumMaster acts as a facilitator and problem-solver for the team's working processes, while the Product Owner determines, explains, and prioritizes project requirements from the business's or client's point of view.

summer 2014 semester was only six weeks long, so sprints were held weekly.

The instructor chose to design the course this way in anticipation of three particular benefits: (1) that the students would work on their projects at a sustainable pace throughout the semester instead of "cramming" in the few final weeks, (2) that short, frequent iterations would create more opportunities for students to reflect and learn about the technology as well as the client's needs, and (3) that frequent iterations would give the instructor greater visibility into the progress and impediments faced by student teams. Anecdotally, each of these benefits was realized to some degree.

3.2. Data Collection

Several types of data were collected as part of the course, for the general purposes of understanding the student experience and improving course design. These data collection points included:

1. An initial questionnaire asking students about their skill levels and project assignment preferences.
2. Four questionnaires completed by students during sprint retrospective meetings, identifying success factors and impediments encountered during the preceding sprint.
3. Weekly entries on project blogs by individual students.
4. Project reports written by students, summarizing their experiences of the capstone project.
5. An exit survey for clients, asking for qualitative and quantitative feedback on their experiences working with the student teams.

Following the end of the semester, we contacted those clients who gave us permission to do so and requested interviews. Of the 27 clients[3], 21 completed the exit survey and 13 have granted interviews as of this writing.

For the present research, we drew on data from the client exit surveys, particularly client responses to open-ended questions, and augmented it with semi-structured client interviews.

[3] Although the 22 project teams worked with only 21 client companies, the company which sponsored two projects assigned a different employee to work with each student team, and these two client representatives filled out separate exit surveys.

3.3. Data Analysis

Our investigation focused on (a) clients' initial framing of the envisioned software, (b) clients' initial framing of how the project would unfold, (c) clients' awareness of new constraints or other adjustments made to their frame following the beginning of the course project, and (d) their willingness and ability to adjust to their frames.

We found evidence that an initial framing of the technology project had a persistent effect on the project, in spite of a client becoming aware of additional constraints, supporting our model. We discuss how each construct of our model is measured from the data below. Most of the data was obtained from client interviews. Whenever possible, we interviewed the students to get additional information on the clients' perspectives and behavior.

Initial Frame content: Initial frame content was assessed from the client based on their actions at the beginning of the project, as well as what they learned as the project progressed. We assume that learned frame content was absent at the beginning of the project.

Frame disruption/learning: We asked clients what they learned which might have had an impact on the project, and what they might have done differently had they redone the project.

Frame Resistance to change: To identify resistance to change, we asked clients if the learning had any impact on the project.

*Client Collaboration***:** To capture the extent of customer/client collaboration, we asked clients to report how often they communicated face to face and through email in a survey.

3.4. Findings

3.4.1. Initial Client Frame Content. There was a fairly clean grouping of clients our dataset into two types: entrepreneurs seeking to enhance their small business and employees of larger firms who were seeking to enhance an existing business process. Entrepreneurs had a much stronger business requirement focus, whereas employees, especially those with technical experience had much more clarity and focus on technical requirements.

Knowledge of business requirements: Clients in a majority of the projects had a passionate, detailed

focus about the developed technologies. Many of the clients were very detailed and specific about the design of the product. One team of students confirmed that their team actually had wireframe mockups of the website they desired to build. Other teams reported that their customers were confident in what they wanted, but allowed more flexibility in the execution. For example, a new application for managing employee time clocks could be built either as a native smartphone app or as a mobile website.

Knowledge of Technical Requirements: Technically experienced clients had much more extensive knowledge of the requirements. One client proposed to develop a web scraping tool to keep a list of medical products updated. The technical requirements—which language the web scraping tool was going to be designed and expanded remained constant over the course of the project. Another client had specific recommendations for the styling of a web page used to coordinate training within a large company.

Mentoring Role: In some cases, clients clearly mentioned their beginning with a mentor role, and knew how the students would approach the work. One client performed their own skill assessment survey as part of the initial project planning. Other clients were aware that they should have some role with respect to the project, but did not know what that would be.

Project Goals: Goals for the project varied from looking for students to hire, to developing further knowledge of a prototype to obtaining a useable product at the end of the semester.

3.4.2. Frame Disruption/Learning. Clients were asked about new insights that emerged during the course of the project, and whether these new insights led to requirements changes, both in the questionnaire and in the follow-up interviews.

Student Abilities: The most widely reported learning focused on student abilities and student behaviors. Even clients with more software project experience reported learning that students did not have certain technical abilities, such as the ability to test a landing page. Another client noticed that students were not working well together as a group. One client realized half way through the project that students would not be able to complete the project. Clients who assumed a mentor role did not report these types of disruptions. One client noted that they would become more involved in coaching the students to work together as a team if given the chance to be involved in a second project to help more of the students get involved in the projects.

Clients perceived that students were not initially forthright with their skill sets, perhaps because they wished to challenge themselves by taking on projects that sounded interesting despite being beyond their current skills. One client noted that several teams were interested in her company's project, despite her belief that it was a more complex project than several of the others.

Technical Constraints: Several of the entrepreneurs and small business owners learned about technical constraints over the course of the project. One owner had to adjust the contents which would be displayed on an iPad Mini since the screens were too small. In the end a full sized iPad was used. In other cases, the client was forced to realize that parts of the technology would require much more effort than had previously been previously envisioned. One client felt that he learned that he needed to develop a stronger sense of prioritization of business requirements.

Project Mechanics: One client reported learning that they needed to be better at making design decisions on the fly, as well as the need to prioritize requirements.

External Project Changes: Two of the clients reported needing to respond to external changes. One client, for example, split their startup into two companies, a nonprofit and a for-profit business. Another client had marketing request a branding change, which required significant changes within the project.

3.4.3. Frame Resistance to Change. Clients were asked an open-ended question about how well they felt the goals of their project had been met, and asked to give quantitative scores to the students that would impact their grades. In follow-up interviews, they were asked if they and the developers had different expectations about the projects, a question that elicited comments on the overall performance of the teams.

In almost every case, the chief regret of the clients was that they did not know from the beginning what the students were capable of. If they'd known sooner, several of them believed, they could have better tailored their requirements to the students' abilities, and raised the bar for project success.

What is noteworthy about the interview data is that in no case did a client report significantly

adjusting requirements in order to take advantage of student abilities after the project began. Although some clients were pleasantly surprised by student abilities in, for example, the digital marketing area, they were not able to adjust requirements to gain more benefits from these skills. This is despite the fact that clarity about students' capabilities came about fairly early, generally by the end of the first of four sprints.

Another common remark was that the clients didn't fully understand the focus or intent of the capstone course. They were unsure whether they were expected to provide a challenging learning experience, or to offer a simpler project that students would be better able to complete. This doubt may have prevented some clients from adjusting their projects mid-semester. No clients reported assuming a mentoring role who did not assume one at the beginning.

Several clients reported that they had been able to adjust their requirements to meet technical constraints, but this was often too late to avoid substantial rework. The clients who had technical experience with software projects did not report any disruptions or inability to adapt to changing requirements.

Perhaps a reason for a lack of framing adjustment was that most clients were satisfied with the work that the students provided. Overall, in all but a few cases the clients gave very high marks to the students. Even when reporting that the teams did not have the technical skills that were expected, they were pleased by the students' hard work and insights. For example, one client, expecting web development work, was pleasantly surprised that the students had expertise to contribute to social media and digital marketing strategy as well.

3.4.4. Agile Communication. We verified the extent of client communication to assure that this was not the cause of a lack of frame disruption or of resistance to change. Clients were asked how often they interacted with student teams via any medium, how often they met face-to-face, what was discussed during meetings, and how thoroughly they discussed the "backlog" or initial set of requirements over the course of the three-month projects. In the follow-up interview, clients were asked, "What were the most important considerations you had in mind about the project's scope, design, as well as how to work with the team?"

All of the clients reported meeting face-to-face with the students at least four times per semester or at least once per sprint. In almost every case, clients reported that they were initially unaware of the development team's skills and capabilities, but discovered this during or after the first sprint. Several clients commented that they would have been able to better utilize the student teams if they had been made aware of their abilities earlier.

4. Summary of Findings

A fairly consistent story emerged from this data set: clients were initially unaware of the skill sets of the student development teams. In almost every case, they learned within the first sprint about the shortfalls of the students' skills and some surprising strengths.

Clients were generally satisfied with the work performed by the students. However, they did feel that the student teams could have been utilized more effectively – they could have provided more work, better suited to the skill sets of the teams, if they had known these at the beginning. The clients who did not report this problem seemed to be the ones who either had simple projects or assumed a low skill set on the part of the students.

The interesting thing is that, even though the clients learned about the teams' skill sets fairly early, at least by the half-way point, it was too late to make a course correction. Even though each client reports collaborating with the students at least once per sprint, it appears they decided to stick with their initial plans. This inertia may or may not have to do with clients' doubts about their freedom, as part of the capstone course, to change the project midway through the semester.

Adjusting one's frame to account for this potential source of setbacks appears to have occurred in spite of an iterative, collaborative approach to developing requirements. This appears to have happened because the clients did not appear to feel responsible to assume their role as mentors if they hadn't done so before the beginning of the project.

5. Discussion

This research investigated the extent to which clients are able to make adjustments to an initial framing of an Agile software development project in a short time frame. We find that the data from our study of student capstone projects supports our claim that client frames can prohibit adaptability when the client is unable or unwilling to make adjustments to their framing of the project.

We argued that the concept of technology project frame disruption could be useful in capturing the

learning by clients and developers at the beginning of such a project, and proposed a model of client frame disruption and adjustment in software development environments.

Perhaps the most significant finding from the client interviews is that Agile project clients came to learn about the range of the development teams' abilities and did not adapt to them in spite of the potential impact on the project. Despite the fact that this discovery occurred very early, clients were unprepared to take advantage of this learning by adjusting project requirements. This raises a new question: why? And what can we learn from this that applies to Agile projects more generally?

This research context has a number of limitations, and identifying them here may help us to improve the research design of future studies. First, these projects were completed as part of a capstone course, and it is possible that clients chose not to adjust project requirements because they did not know what the professor expected of them as participants.

Furthermore, the student teams were not highly experienced professionals, but rather needed to learn the technologies (for web development, mobile development) on the job. This may differ from the broader population of Agile ISD projects, where it is generally the developers who can teach the clients about the affordances and constraints of alternative technologies. This limitation could explain why we did not hear reports of clients' technology frames being disrupted – they did not have expert technologists on the developer team to disrupt them.

Nevertheless, this research has turned up two important discoveries: first, that the lack of early communication about developers' abilities was a stumbling block to most clients in this research context, and second, that clients were unable to adjust requirements after learning about the developers' abilities. Whether this inability to adjust requirements was due to role confusion on the part of the clients, to an a priori inflexibility of the projects, or to some other problem, this research context has the potential to shed light on a reason why, despite the presence of good Agile communication and iteration, course changes failed to occur.

6. Ongoing Research

One of the authors of this paper will continue to teach the capstone course for information systems undergraduates in the coming year: fall 2014 and spring 2015. Armed with the insights of this opportunistic study, which may be considered a pilot study, we can re-design data collection to better capture the technology frames, and technology frame disruptions, which we believe may hold the secret to understanding client learning on short-term and early-stage Agile engagements.

We found several patterns of client frame content, frame disruption, and adaptation which we did not collect enough data to conclusively support. Clients with additional technical expertise appeared to have no major disruption related to discovering technological constraints whereas clients without this expertise had major setback resulting from discovering technical constraints. Those who assumed the role of a mentor appeared to have no difficulty working with student's abilities.

7. Future Directions

The challenges faced by clients and developers in short-term and early-stage collaborations are real, and have been the subject of little study. Given the assumption of sufficient time to "work out the kinks", most writers on the subject of Agile ISD methodology overlook these challenges in favor of focusing on more perennial challenges to agility. We call upon researchers to bring the focus back on to the initial clash of viewpoints and the initial process by which clients and developers learn to collaborate toward the achievement of a new goal. This research has begun, and will continue, to investigate this process using the concept of technology frame disruption.

Our research design takes advantage of a convenient population of projects that are similar in terms of scope, calendar duration, team size and developer experience, and which can be studied longitudinally. The limitation is that these student teams may not adequately represent the characteristics of other Agile ISD teams worldwide. Thus, there is a great need for further research that targets diverse Agile ISD teams in the business world, perhaps by large-scale survey.

Finally, our initial research has identified a recurring case in which, despite the use of Agile processes and the occurrence of good and frequent communication, product owners were unable to adjust requirements to take advantage of developer capabilities. Research is needed to understand what makes this occur, and in what contexts it may occur.

8. References

[1] K. Conboy, "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development", Information Systems Research 20(3), 2009, pp. 329-354.

[2] G. Lee and W. Xia, "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility", MIS Quarterly 34(1), 2010, pp. 87-114.

[3] R. Vidgen and X. Wang, "Coevolving Systems and the Organization of Agile Software Development", Information Systems Research 20(3), 2009, pp. 355-376.

[4] R. Hoda, J. Noble, and S. Marshall, "The impact of inadequate customer collaboration on self-organizing Agile teams", Information and Software Technology 53, 2011, pp. 521-534.

[5] E. J. Davidson, "Technology Frames and Framing: A Socio-Cognitive Investigation of Requirements Determination", MIS Quarterly 26(4), 2002, pp. 329-358.

[6] B. Azad, and S. Faraj, "Making E-Government Systems Workable: Exploring the Evolution of Frames", Journal of Strategic Information Systems Management 17(2), 2008, pp. 75-98.

[7] R. Appan, and G. J. Browne, "The Impact of Analyst-Induced Misinformation on the Requirements Elicitation Process", MIS Quarterly 36(1), 2012, pp. 85-106.
.

[8] G. M. Marakas, and J. J. Elam, "Semantic Structuring in Analyst Acquisition and Representation of Facts in Requirements Analysis", Information Systems Research 9(1), 1998, pp. 37-63.

[9] J. W. Moody, J. E. Blanton, and P. H. Cheney, "A Theoretically Grounded Approach to Assist Memory Recall During Information Requirements Determination", Journal of Management Informantion Systems 15(1), 1998, pp. 79-98.

[10] M. G. Pitts, and G. J. Browne, "Improving Requirements Elicitation: An Empirical Investigation of Procedural Prompts", Information Systems Journal 17(1), 2007, pp. 89-110.

[11] R. W. Zmud, W. P. Anthony, and R. M. Stair Jr., "The Use of Mental Imagery to Facilitate Information Identification in Requirements Analysis", Journal of Management Information Systems 9(4), 1993, pp. 175-191.

[12] P. Barr, "Adapting to Unfamiliar Environmental Events: A Look at the Evolution of Interpretation and Its Role in Strategic Change", Organization Science 9(6), 1998, pp. 644-670.

[13] J. Bartunek, "Changing Interpretive Schemes and Organizational Restructuring: The Example of a Religious Order", Administrative Science Quarterly (29), 1984, pp. 355-372.

[14] D. Gioia, "Symbols, Scripts, and Sensemaking: Creating Meaning in the Organizational Experience", in The Thinking Organization, H. Sims Jr. and D. Gioia and Associates (eds.). Jossey-Bass, San Francisco, 1986, pp. 49-74.

[15] G. Salaway, "An Organizational Learning Approach to Information Systems Development", MIS Quarterly 11(2), 1987, pp. 245-264.

[16] W. J. Orlikowski, and D. C. Gash, "Technological Frames: Making Sense of Information Technology in Organizations", ACM Transactions on Information Systems 12(2), 1994, pp. 174-207.

[17] J. P. Walsh, "Managerial and Organizational Congition: Notes from a Trip Down Memory Lane", Organization Science 6(3), 1995, pp. 280-321.

[18] P. M. Leonardi, "Innovation Blindness: Culture, Frames, and Cross-Boundary Problem Construction in the Development of New Technology Concepts", Organization Science 22(2), 2011, pp. 347-369.

[19] J. Holstrom, and S. Sawyer, "Requirements Engineering Blinders: Exploring Information Systems Developers' Black-Boxing of the Emergent Character of Requirements", European Journal of Information Systems (20), 2010, pp. 34-47.

[20] K. Conboy, and B. Fitzgerald, B. "Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion", ACM Transactions on Software Engineering Methodology 20(1), 2010, pp. 1-30.

[21] B. Curtis, H. Krasner, and N. Iscoe, "A Field Study of the Software Design Process of Large Systems", Communications of the ACM 31(11), 1988, pp. 1268-1287.

[22] N. A. M. Maiden, and G. Rugg, "Acre: Selecting Methods for Requirements Acquisition", Software Engineering Journal 11(3), 1996, pp. 183-192.

[23] D. B. Walz, J. J. Elam, and B. Curtis, "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration", Communications of the ACM 36(10), 1993, pp. 63-77.

[24] J. Coughlan, M. Lycett, and R. D. Macredie, "Communication Issues in Requirements Elicitation: A Content Analysis of Stakeholder Experiences", Information and Software Technology 45(8), 2003, pp. 525-537.

[25] M. Limayem, and G. DeSanctis, "Providing Decisional Guidance for Multicriteria Decision Making in

Groups", Information Systems Research 11(4), 2000, pp. 386-401.

[26] S. Gasson, "A Genealogical Study of Boundary-Spanning IS Design", European Journal of Information Systems (15), 2006, pp. 26-41.

[27] S. Chakraborty, S. Sarker, and S. Sarker, "An Exploration into the Process of Requirements Elicitation: A Grounded Approach", Journal of the Association for Information Systems 11(4), 2010, pp. 212-249.

[28] A. Tiwana, "An Empirical Study of the Effect of Knowledge Integration on Software Development Performance", Information and Software Technology 46(13), 2004, pp. 899-906.

[29] M. C. Charaf, C. Rosenkranz, and R. Holten, "The Emergence of Shared Understanding: Applying Functional Pragmatics to Study the Requirements Development Process", Information Systems Journal 23(2), 2013, pp. 115-135.

[30] L. M. Maruping, V. Venkatesh, and R. Agarwal, "A Control Theory Perspective on Agile Methodology Use an Changing User Requirements", Information Systems Research 20(3), 2009, pp. 377-399.

[31] A. Elssamadisy, and G. Schalliol, "Recognizing and Responding to 'Bad Smells' in Extreme Programming", Proceedings of the 24th International Conference on Software Engineering, 2002, pp. 617-622.

[32] R. L. Purvis, and V. Sambamurthy, "Factors Influencing the Gap between User-Designer Perceptions of the Systems Analysis Process", Journal of Information Technology Management 8(3,4), 1997, pp. 21-31.