

Fostering effective inter-team knowledge sharing in agile software development

Viviane Santos · Alfredo Goldman ·
Cleudson R. B. de Souza

Published online: 10 May 2014
© Springer Science+Business Media New York 2014

Abstract Agile methods have different practices that ultimately foster intra-team knowledge sharing (KS). However, they do not provide recommendations about coping with inter-team KS. Supporting inter-team KS is fundamental in scaling agility across the entire organization, and is regarded as the new horizon for agile software development. This study seeks to understand inter-team knowledge sharing effectiveness in agile software development organizations. Using grounded theory, we analyzed data from four Brazilian organizations and also from an expert in agile methods implementation. We observed that the companies employ different work practices that allow knowledge sharing to occur across team boundaries. Our main contribution is a conceptual model that explains how effective knowledge sharing across agile teams depends on applying purposeful practices, along with organizational conditions and stimuli. This understanding represents an innovative focus within the context of agile methods and suggests what one needs to take into account when striving on this path. Yet, it yields opportunities for further studies in refining and extending the model to other organizational contexts. Inter-team knowledge sharing reflects the way agile software development organizations are coping with enterprise agility and the way they consider knowledge as a resource for competitiveness.

Keywords Agile methods · Inter-team knowledge sharing effectiveness · Practices · Organizational conditions · Stimuli · Conceptual model

Communicated by: Jeffrey C. Carver

V. Santos (✉) · A. Goldman
Department of Computer Science, University of São Paulo, Rua do Matão, 1010, Cidade Universitária,
São Paulo, SP, 05508-090, Brazil
e-mail: vsantos@ime.usp.br

A. Goldman
e-mail: gold@ime.usp.br

C. R. B. de Souza
Vale Institute of Technology and Federal University of Pará, Tv. Boaventura da Silva, 955, Nazaré,
Belém, Pará, 66035-090, Brazil
e-mail: cleudson.desouza@acm.org

1 Introduction

Agile software development (ASD) is a lightweight paradigm for software development that emerged in response to the limitations of the prescriptive methodologies, e.g., the waterfall approach (Sommerville 2010). This software development paradigm can be very effective, since its focus is on empowering the team in achieving its goals, embracing changes, delivering working software in iterative and incremental cycles, and taking into account the human aspects of this endeavor (Dingsøy et al. 2012).

From a Knowledge Management (KM) perspective, the focus of ASD approaches is on sharing tacit knowledge within the team. According to Cummings (2004), Knowledge Sharing (KS) is the provision of task information and know-how to a person, so that (s)he can collaborate with others to solve problems, develop new ideas, or implement policies or procedures. KS can occur through tools and repositories (codification strategy), or via face-to-face communication among the social actors (personalization strategy) (Hansen et al. 1999; Sveiby 2004). Knowledge sharing is one of the most important knowledge management processes, which gradually improves an organization's production system and its elements. As a result, knowledge sharing is closely related to long-term organizational performance and competitiveness (Du 2007).

In agile software development, intra-team knowledge sharing is regarded as important for accomplishing specific project tasks and for offering the opportunity to discover creative means to improve the organization's competitiveness. The agile culture helps nurturing a natural environment for intra-team knowledge sharing, which is achieved by emphasizing face-to-face conversations through agile practices and by using minimal documentation (Boden et al. 2009). Thus, agile methods are in line with KM personalization strategies.

Furthermore, the agile philosophy supports the changes needed for a KM endeavor, which are critical to sustain a knowledge sharing culture (Santos et al. 2011; Levy and Hazzan 2009; Nerur et al. 2005). Agile values and principles foster changes in team members' attitudes and strengthen their relationships. These changes happen as a result of greater trust and better communication and transparency in the relationships among team members and customers. Likewise, the agile philosophy provides less imposition by managers, greater sense of commitment, reciprocity, responsibility, and freedom of expression. Also, this approach promotes a better understanding of the work processes, and decreases individual dependence through collective code ownership, cross-functional and self-organizing teams (Karlsen et al. 2011; Qureshi and Kashif 2009).

Several authors have pointed out that agile methods facilitate knowledge sharing within the team, but offer little explicit support for inter-team knowledge sharing and learning (Chau et al. 2003; Holz et al. 2003; Chau and Maurer 2010; Holz and Melnik 2004; Chau 2005; Karlsen et al. 2011; Neves et al. 2011). Kettunen and Laanti (2008) also argue that in agile methods the tacit knowledge is appreciated and the free flow of knowledge is assumed, but no special emphasis is given to organizational knowledge management. They make a comparison with agile manufacturing processes, as these highlight this concern, but do not provide focused guidance on how to accomplish knowledge management. Similarly, Pikkarainen et al. (2008) report that agile practices do not recommend mechanisms of communication or knowledge sharing among agile teams and their stakeholders.

Thus, the excessive focus on the product and on the delivery of value to customer, lack of knowledge sharing practices, or time-pressure can make inter-team knowledge sharing difficult and, further, may lead teams to repeat past mistakes, rather than learn from experiences (Karlsen et al. 2011). In other words, agile methods may hinder the effective creation

and sustenance of organizational knowledge (Karlsen et al. 2011; Bjørnson 2008; Chau and Maurer 2004; Kähkönen 2004).

Given the greater focus on KM personalization strategies in agile software development, the research question guiding this enquiry is: *How organizations achieve knowledge sharing effectiveness across agile teams using personalization strategies?* To answer this question we conducted a Grounded Theory (GT) study (Glaser and Strauss 1967) in four Brazilian agile software development organizations that adopt inter-team knowledge sharing practices. We also validated our results with an expert in agile methods implementation.

The contribution of this paper is a conceptual framework that takes into account organizational conditions, stimuli and practices for effective inter-team knowledge sharing in agile contexts. This framework helps explaining how the influencing factors impact the knowledge sharing process in agile software development organizations. As a consequence, collective knowledge can be more easily scaled to the organizational level (Shalloway and Beaver 2009). This framework is our theory and provides explanations on inter-team knowledge sharing within the investigated agile organizations. We recognize that this theory is somewhat limited to the presented contexts and requires further study in other organizations in order to elaborate upon the theory towards generalization. Nevertheless, this theory may help practitioners understand the inter-team KS initiatives in their own organizations, or at least stimulate them to think about this theory within their contexts.

The rest of this paper is organized as follows: Section 2 presents a literature review of knowledge sharing approaches in ASD; Section 3 describes the settings, while Section 4 explains the research method adopted. Sections 5 and 6 report our main results: inter-team knowledge sharing practices, influencing factors and effectiveness. Section 7 contains a discussion on the results, while Section 8 discusses the limitations of our study. Finally, Section 9 presents the final remarks and possibilities for further work.

2 Inter-team Knowledge Sharing in Agile Software Development

As described in the previous section, the focus of our research is on inter-team knowledge sharing in agile software development organizations. Therefore, our first step was to conduct a broad literature review on this topic, as a way to understand the characteristics of the previous studies of this research field. This section presents a summary of these results and we aim to make it available to the community and other researchers.

After executing search strategies in databases acknowledged by the academic community, we concluded that inter-team KS in ASD is still an unsolved issue, i.e. the reported studies with methodological rigor that seek theories to explain this phenomenon are still only a few.

Table 1 lists the relevant retrieved studies from a systematic literature review, which are grouped into the two types of knowledge management strategies (Hansen et al. 1999; Sveiby 2004). The codification strategy focuses on making knowledge explicit through tools and repositories. On the other hand, the personalization strategy concentrates on fostering interaction among people, since knowledge sharing will happen during these interactions. Further, we briefly discuss these studies.

2.1 Studies Regarded as Codification Strategy

We retrieved some interesting studies that aim to make knowledge explicit; these are described below. Stettina et al. (2012) stated that particular care should be taken to add

Table 1 Relevant studies retrieved from literature review

Type of KM strategy	Studies	Total
Codification	(Stettina et al. 2012; Kavitha 2011; Wang et al. 2010; Rech and Bogner 2010; Maalej and Happel 2008; Chau 2005; Chau and Maurer 2004; Holz and Maurer 2003; Holz and Schafer 2003)	9
Personalization	(Maranzato et al. 2011; Faegri et al. 2010; Kettunen 2010; Faegri 2009; Srinivasan and Lundqvist 2009; Conboy 2009; Cabral et al. 2009; Bjørnson 2008; Kettunen and Laanti 2008; Mestad et al. 2007; Keplinger 2007; Salo and Abrahamsson 2007; Lyytinen and Rose 2007; Pikkarainen et al. 2005; Kähkönen 2004; Desouza 2003)	16
Total		25

documentation practices to agile teams. They learned that codification of internal development knowledge should be a non-intrusive task.

Holz and Maurer (2003) developed PRIME, a system that uses search strategies and information retrieval to capture and distribute knowledge about specific tasks from available information sources that contain information which then can be potentially used to meet the needs of other teams. This system was applied to Scrum teams in academic environments to identify and support communities of practice, and to reduce the problem of maintaining knowledge repositories. The proposal of these authors was still in preliminary stage and there was only one more article about the evolution of this tool (Holz and Schafer 2003), since then there has been no more publications about it, new versions of the tool have apparently discontinued.

The MASE environment (Chau 2005; Chau and Maurer 2004) is a web-based collaboration and knowledge sharing tool for agile teams. It was developed to provide a means to facilitate knowledge sharing and cooperation among distributed teams, to undertake synchronous and asynchronous work, and to facilitate the creation of communities of practice (CoP) as a way to foster organizational learning. After its development, this tool was not applied in an organizational setting and was ceased by its research department.

Maalej and Happel (2008) proposed a framework based on ontologies to capture, access and share experiences of developers on a decentralized context. This framework captures the developers' interaction with related artifacts and provides an annotation-based approach Wiki that triggers knowledge capture. Similarly, Rech and Bogner (2010) developed a knowledge management system focused on people, which is integrated with RISE (a reuse tool in software engineering) and uses Wiki-based semantic for agile software development environments.

Recently, Kavitha (2011) proposed another framework to generate organizational learning through capturing, storing and disseminating tacit knowledge gathered from Extreme Programming (XP) practices of Pair Programming (PP) and Pair Rotation within teams. This author considered using collaborative tools to help capturing experiences and structuring them in idea maps and forums in order to store collective tacit knowledge in a repository. Though it seems to be an interesting proposition, the author has not yet reported more details about the framework and its validation in a real setting.

After performing a critical analysis of several tools that enabled distributed agile planning, Wang et al. (2010) offer a series of recommendations for designers of these tools, including the need to support synchronous interactions, verbal communication, creation of

shared informative workspaces, visualization of interactions of distributed users, ubiquitous project planning and information reuse from other planning tools.

Our conclusions about the studies involving the codification strategies are that the proposed tools were not adopted and evaluated by companies, and most of them have been discontinued. The tools referred here have advantages such as enabling collaboration among co-located or distributed teams, but still need to be properly evaluated.

A problem to be solved is related to the projects' time pressure and low prioritization of extensive documentation in agile environments. These issues might imply in low quality of the repositories' content, and difficulty in finding and maintaining information. There is a need to improve content reuse through consistent structures with relevant and updated information. As some of the cited authors explain, users often fail to provide helpful data or simply do not care about complex sets of data, which is in line with what (Brössler 1999) states that, in general, software team members usually are not aware of which knowledge is relevant to share.

2.2 Studies Regarded as Personalization Strategy

The first personalization strategy for inter-team knowledge sharing identified in the literature is the Scrum of Scrums. Scrum of Scrums is a short meeting to coordinate cross-project sub-teams. These meetings involve senior people and common technical areas (Schwaber and Beedle 2002) to solve critical problems or to discuss projects integration and overlapping areas (Srinivasan and Lundqvist 2009; Lindvall et al. 2002). However, to work well, project members, especially, Product Owners (POs), and Scrum Masters need to have a broad view of the interrelated projects.

Desouza (2003) argues that effective knowledge sharing primarily requires getting people motivated to talk and share their know-how. As knowledge is generated by people at the individual level, he proposes, instead of tools, spaces, such as coffee rooms, used to promote knowledge sharing.

Communities of practice (CoPs) are a mechanism for sharing knowledge in self-organizing informal communities (Wenger et al. 2002; Brown and Duguid 1991). Kähkönen (2004) and Mestad et al. (2007) reported the usage of CoPs in agile environments. These authors emphasize the dynamic and human centric approach of agile methods, in which CoPs need to be adapted to a flexible model, like dynamic skill circles with voluntary and free participation.

Project members rotation is a common practice of transferring workers from one project team to another for knowledge leveling and redundancy (Fægri 2009; Fægri et al. 2010). This practice is still challenging, since it incurs a collective cost that must be amortized and legitimized by the organization, and yet can expose implicit organizational values. Likewise, Birds of a feather (BoF) sessions are informal discussion groups, usually held on agile conferences (Conboy and Fitzgerald 2010). BoFs have been adapted and tailored to work in agile contexts for sharing knowledge (Santos et al. 2012).

Open fishbowl sessions¹ are used to discuss shared interest topics within large groups. This practice was firstly held on OOPSLA (Object-Oriented Programming, Systems, Languages & Applications) conferences (Fraser et al. 2006) and is described by Santos et al. (2012). It allows disciplined interaction among participants to solve problems, and discussions on how to improve or reuse solutions, and raise feedback.

¹[http://en.wikipedia.org/wiki/Fishbowl_\(conversation\)](http://en.wikipedia.org/wiki/Fishbowl_(conversation))

A study published by software development practitioners of the Research and Development department at Universo Online (UOL) (Maranzato et al. 2011) reported that they share experience among teams through Scrum of Scrums, and through rotation of people among projects (during or after project completion). However, to add value to teams, these practices require more integration among Scrum Masters and Product Owners (POs) as well as a wide view of the products. Besides, the authors report the adoption of technical presentations about training or conferences that members of the department have attended. They also report on relevant characteristics or best practices that they consider important to share with everyone.

Some studies relate the concepts of agile approach to Organizational Learning (OL). As part of our research, we have retrieved one study analyzing the Argyris' double-loop learning approach (Kettunen 2010), two studies that are investigating the dilemma of balancing between the exploitation and the reuse of knowledge in the organization (Keplinger 2007; Lyytinen and Rose 2007), and two studies describing the importance of organizational learning to the agile approach (Salo and Abrahamsson 2007; Pikkariainen et al. 2005). These latter studies emphasize the need for greater support for the process itself to improve long-term learning. Furthermore, other studies indicate the need for future research between organizational learning and agility (Conboy 2009; Cabral et al. 2009).

Our conclusions about the studies about the personalization strategies are that there is a need to have more studies on this topic, because most of them present approaches adopted by agile teams in only one company. We still lack understanding on how to apply these practices effectively.

As the codification strategies conducted so far in agile contexts are still a challenge and agile methods rely on the personalization strategies to spread tacit knowledge within the team, scaling personalization strategies across teams seem to be more sustainable in agile contexts, than the codification strategies. Because of this aspect, our study aims to provide an emergent theory in this research area, grounded in data gathered from four agile organizations that are described as follows.

3 Settings

In this section we describe our research settings, i.e., the main criteria for selecting the organizations as well as the studied organizations.

The criteria for selecting Brazilian organizations to participate in this qualitative study were the following: (1) the organization should have more than one agile team to allow for exploring cross-team knowledge sharing; (2) agile method(s) should be adopted for at least one year to benefit from the agile philosophy that empowers intra-team tacit knowledge sharing; and (3) the organization should apply practices for knowledge sharing across teams.

We also took into account the convenience and availability of the organizations' context for answering the research question, the congenial relationships established between the organizations and the university/researchers, and the potential for the organization to learn from the research context, despite the fact that the data was collected at an early stage of the research.

As the next step, we selected experienced agile software development organizations from different business domains and sizes that employ inter-team knowledge sharing practices. It is important to highlight that the companies participating in this research allowed mentioning their names because the topic under study was openly considered important by them. Moreover, these companies are interested in sharing their own information and have already

published their names in previous studies. Thus, presenting their own names would help the community follow their data. In Table 2 we describe some of the companies' relevant details. An expert in agile methods implementation was also interviewed for enhancing data source triangulation (Runeson and Höst 2009), since it provided a broader picture of inter-team knowledge sharing.

Universo Online (UOL) is a large-sized company. The studied department has implemented Scrum (Schwaber 2004) complemented with other practices. They have also tried to adopt some XP practices, such as continuous integration (CI), test-driven development (TDD) (Beck 2002) and PP. However, they still need to overcome some organizational barriers, such as top management and developers convincement, investment in infrastructure and training, and they need to change the development culture.

Apontador is a medium-sized company with interrelated agile projects. They have shared members across the teams, e.g. Scrum Master, senior developer, QA (Quality Assurance), web designers, and DBA (Database Administrator). In 2010, they started adopting some XP practices, like PP, CI, user stories, small releases, metaphor, collective ownership, coding standards, simple design, refactoring, testing, 40-hour workweek, and on-site customer.

Caelum is a medium-sized company with 30 developers working in open spaces. Learning and teaching is fundamental to the company's business domain (Aniche and Silveira 2011), as presented in Table 2. Their actual focus is on XP practices such as PP, CI, automated tests, simple design, and refactoring. TDD and small releases are not yet emphasized in the company software development department.

ThoughtWorks is a global large-sized company. Their main objective is not only to deliver products to their customers, but also to spread the agile culture to other software companies. In 2009, ThoughtWorks opened a medium-sized branch office in Brazil. As in the headquarters, TW-Brazil adapts agile methods in five large teams, one being composed of five sub-teams, each working for the same client. Developers work in pairs in most working tasks (an average of seven hours a day) within the team (local) and with distant teams (remote). Their working tasks include PP, writing e-mails, preparing internal or external presentations, book translation and even in tasks that are not related to work.

The agile expert name is Heitor Roriz Filho, who is a specialist in agile methods (agile coach and trainer) since 2004 (Corbucci et al. 2011) at Massimus. Given his experience in several software development organizations and also in knowledge management (Silva et al. 2010) for about eight years, his point of view was important to enhance reliability and validity of this research (Merriam 2002; Altheide and Johnson 1994), as he assessed whether the results gathered from the organizations could be found in other contexts as well. This assessment is named as *Member checking* (Schwandt 2007). It allowed us to compare our interpretations with those of the research subjects to establish the level of correspondence between the two sets.

4 Research Methods

Our aim in this research was to explore effectiveness in inter-team knowledge sharing and obtain richer and more informative results related to this phenomenon in agile software development organizations. Thus, we chose grounded theory (Glaser and Strauss 1967) to help us to answer the research question: *How organizations achieve knowledge sharing effectiveness across agile teams using personalization strategies?*

Table 2 Selected agile organizations

	UOL	Apointador	Caelum	TW-Brazil
Total company size	> 1000	80	>60	> 100
Business domain	Content portal	Local search platform	Teaching and software development	Software development and consulting
Agile methods	Scrum and XP (few practices)	Scrum and XP (several practices)	Adaptation of Scrum, XP and Kanban	Adaptation of Scrum, XP and Kanban
Experience	4 years	3 years	6 years	>10 years
Agile teams	8	3	4	5
Type of delivery	3 week iterations	1 week iteration	Continuous flow	2 week iterations

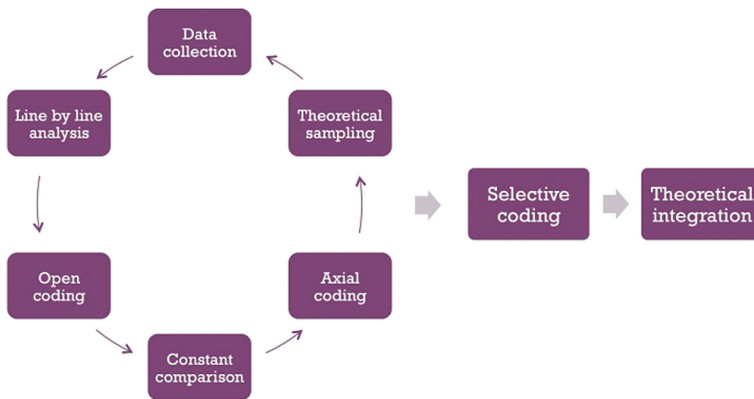


Fig. 1 Iterative research process

We believe that the GT method is appropriate to this study, because given the social nature of KS, which takes place in an organizational context, it required in-depth examinations to capture specific issues and human behavior. Likewise, the GT method of Corbin and Strauss (2007) offers detailed procedures for data analysis (coding techniques) that allows for the emergence of original and rich findings grounded in the data.

We undertook this study in four iterations of data collection and analysis. All recorded interviews and lectures were transcribed, while observations and informal conversations were documented in field notes, totaling 302 pages of evidence. The observations occurred from December 2010 to July 2011 and lasted from 15 min to 4 h. We integrated our field notes to produce detailed records. Some of the relevant quotes are presented in this paper. All the data collection was performed in Portuguese, so we translated our results to English.

Figure 1 depicts the iterative process within and across the selected organizations. We first began to collect data from each case. For each data collection phase, we developed a research protocol to guide the data collection process, considering previous theoretical sampling² (Corbin and Strauss 2007), where it has been available.

We analyzed raw data line by line to identify units of meaning and coded them into *conceptual categories* (concepts) using the open coding technique (Corbin and Strauss 2007). The constant comparison method was employed with the raw data and the identified categories to suggest relationships among the concepts, the so-called *notional categories* (Corbin and Strauss 2007). The category names were based on the terminology used by the informants and on the terms brought by the researchers. This analysis allowed us to develop preliminary theoretical propositions and memos.

This cyclical process was applied as a way to perform GT's *theoretical sampling*. This process helped us to follow through onto the next iteration with more sense of direction, subsequent questions, and listen and observe in more sensitive ways. We closely examined the internal consistency of the categories and properties³ being conceived. The emergence of new concepts that did not fit into the existing set of categories and properties forced us to question the emergent model and follow up with additional data collection.

²Theoretical sampling is the process of choosing new research sites or cases to compare with ones that have already been studied to deepen the understanding and to incorporate all the relevant data into the analysis of the research topic.

³Characteristics of an object, event, or action.

Table 3 Iterations summary

	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Inner questions	How does OL occur in agile organizations?	Which practices are adopted to share knowledge across agile teams?	How do teams achieve effectiveness in inter-team knowledge sharing?	Confirmations and refinements of influencing factors
Period	Nov-Dec/2010 and Jan/2011	Mar-May/2011	Jun-Aug/2011	Sept-Nov/2011
Participants	1 company + 1 consultant	2 companies	4 companies + 1 consultant	4 companies + 1 consultant
Number of interviews	2 group interviews + 1 individual interview	1 interview	3 interviews + 3 questionnaires	12 questionnaires
Number of observations	1 observation	6 observations	16 observations	
Other sources	2 lectures + 2 papers	Informal chats	1 paper + 1 lecture	2 feedback reports + 4 feedback presentations
No. of pages of evidence	76	31	143	52

Table 4 Sources of evidence collected from iteration one

Organization	Interviews	Direct observations	Others
UOL	1st semi-structured group interview: Scrum master (1) and executive (1). 2nd semi-structured group interview: Scrum master (1) and executive (1). The interviews resulted in 24 pages.	1 Mega Daily meeting (about 20 people). The observation was documented in field notes, resulting in 4 pages.	2 Lectures: Executive (1) and Scrum master (1). The lectures resulted in 12 pages. Document: Paper #1 about the OL process triggered with Scrum implementation (Santos et al. 2011), 6 pages. Document: Paper #2 about the analysis of 'Ba' during Scrum process (Silva et al. 2010) in Portuguese, 15 pages.
Massimus Consulting	1st semi-structured interview: specialist (1). The interview resulted in 15 pages.	N/A	

Table 5 Sources of evidence collected from iteration two

Organization	Interviews	Direct observations	Others
UOL	N/A	2 Mega Daily meetings (about 30 people) and 1 Mega Planning meeting (about 30 people) and informal communication. The 3 observations were documented in field notes, including detailed field notes record, which resulted in 9 pages.	N/A
Apontador	1st semi-structured group interview: Scrum master (1) and R&D manager (1). The interview resulted in 9 pages.	1 review meeting(20), 1 planning meeting(16), 1 retrospective(16), other observations, such as informal communication and observation of coding and informative workspace. The 3 observations were documented in field notes, including detailed field notes records, which resulted in 13 pages.	N/A

Table 6 Sources of evidence collected in the third cycle

Organization	Interviews	Direct observations	Others
UOL	3rd semi-structured group interview: Scrum masters (2), product owner (1) and team manager (1). The interview resulted in 14 pages.	2 Mega Daily meetings (30) and 2 Mega Planning meetings (30) and informal communication. The 4 observations were documented in field notes, including detailed field notes record, resulting in 12 pages.	N/A
Apontador	1st semi-structured questionnaire with open questions: Scrum master (1) and R&D manager (1) answered in 3 pages.	1 review meeting (17), 1 planning meeting(14), 1 retrospective(14), 1 Coding Dojo session(12), other observations, such as informal communication and observation of coding and informative workspace. The 4 observations were documented in field notes, including detailed field notes records, resulting in 18 pages.	N/A
Caelum	1st semi-structured group interview with open question: software development consultants/trainers (2). The interview was not recorded, but we documented most important statements in 5 pages.	3 daily stand-up meetings (about 16 people), 23 PP, 1 Brown Bags session with retrospective (17), and other observations, such as informal communication, conversations and observation of coding, testing, informative workspace and training preparation. Observations were documented in field notes, including detailed field notes records, which resulted in 62 pages.	Document: Paper #3 about their learning environment (1) (Aniche and Silveira 2011), 7 pages.
ThoughtWorks (Brazil)	1 semi-structured interview with open questions: software development consultant (1). 1 Semi-structured questionnaire with open questions: software development consultants (2) in 4 pages.	N/A	One lecture : Software development consultants (2). The lecture and the informal conversation were recorded and transcribed in 16 pages.
Massimus	1 semi-structured questionnaire with open questions: specialist (1) in 2 pages.	N/A	N/A

Table 7 Sources of evidence of the data collected in the fourth cycle

Organization	Interviews	Others
UOL	Semi-structured questionnaire: Executive, PO (1), Scrum masters (2) and developers (2) in 12 pages.	Feedback report and presentation: Executive (1), PO (2), Scrum master(1) and developer (1). Notes from presentation: 4 pages
Apontador	Semi-structured questionnaire: Scrum master (1) and developers (5) in 12 pages.	Feedback report and presentation: Scrum master (1), developers (8) and QA/Tester (1). Notes from presentation: 2 pages.
Caelum	Semi-structured questionnaire: software development consultants/trainers (9) in 14 pages.	Feedback presentation: software development consultants/trainers (16). Notes from presentation: 3 pages.
ThoughtWorks (Brazil)	Semi-structured questionnaire: software development consultants in 4 pages.	Feedback presentation: software development consultant (1). Notes from presentation: 1 page.

In addition, we systematically compared cross cases until no new categories or questions about the existing ones were generated. These comparisons helped us to reach theoretical saturation. Then, we developed a systematic refinement and integration of previously generated notional categories into a limited set of final notional categories by linking them conceptually and defining a conditional matrix of dimensional themes (commonalities and differences) (Corbin and Strauss 2007).

The final coding step is termed *selective coding*, in which a core category is identified/created and a central descriptive narrative of the phenomenon under study is developed, relating it to the rest of the set of notional categories. Lastly, we make *theoretical integration* by comparing the resulting model to existing theories in literature. Table 3 sums up each iteration.

‘Iteration one’ was structured around three broad themes: characteristics of agile method(s) adoption towards OL; motivation and barriers of knowledge sharing in wider organizational levels; and approaches applied for generating organizational knowledge. Table 4 details the collected evidence. In the ‘second iteration’, we explored the applied approaches in two different organizational contexts (Table 5).

In the ‘third iteration’, we included two more organizations and another interview with the consultant (Table 6). This iteration ended when no new categories, sub-categories, or questioning of existing ones were generated, resulting in preliminary theoretical construct.

In the fourth iteration, we developed a systematic refinement and integration of previously generated notional categories to define a matrix of dimensional themes. Then, we presented our findings to the organizations in feedback reports or sessions. We tried to involve as many people as possible with different roles. This involvement improved qualitative validity and reliability, as the participants could judge the results. Also, it allowed us

to better understand the object of enquiry by the participants' perspective, which is advocated by (Altheide and Johnson 1994) and (Guba and Lincoln 1989). Any discordance was adjusted until a consensus was reached. Any conflict resolution was managed through discussion meetings.

We also validated our perception on the effectiveness' measure of the adopted practices. Then, new data collection was made with different respondents as presented in Table 7. At this stage, we checked if any data was missing or invalid and if the collected data was sufficient to answer the research question.

Considering the "trustworthiness" of qualitative validity (Guba and Lincoln 1994), we provided detailed description on the research context and the assumptions that were central to the research as a way of enhancing transferability. We checked and rechecked the data and performed peer debriefing by discussing the findings among each other and employing a member's checking (Schwandt 2007; Lincoln and Guba 1985) in order to strengthen credibility. Dependability was enhanced by rigorous theoretical sampling and by presentation of our judgments about potential for bias. We used multiple sources of evidence for theoretical sampling to increase confirmability.

5 Practices for Inter-team Knowledge Sharing

In the studied agile software development organizations, inter-team knowledge sharing is achieved by the adoption of **practices for socializing knowledge**. These practices have a proper **description** and a classification according to specific **purposes**.

We observed that inter-team KS is a relevant concern. Most interviewees declared knowledge is not properly reused in their teams, which means that they often redo work, instead of saving time by reusing or improving existing solutions from other teams. As a manager at UOL depicted, "*We have products in which their models are similarly idealized. However, we implement one using a technology and other using another one [technology] without knowing. Sometimes in a right way, sometimes in the wrong way.*" And another Scrum Master added, "*Or we redo work, one redoes the same thing one did before.*"

Most teams fail in documenting their knowledge, as a member at UOL said, "*We still do not know what to document in the wiki*". So, their personalization strategy are regarded as socialization processes (Berger and Luckmann 1967) and the organizations do not feel vulnerable by not making knowledge explicit, because they apply purposeful practices that are people-focused, crucial for emphasizing interactions and dealing with tacit knowledge throughout the organization.

In this section, we describe the work practices, i.e., the way in which the work is performed in the studied organizations. Before showing the practices, we describe the identified purposes, in which we categorize them.

The Purpose of Identifying Knowledge Owner Interviewees state the need to ease finding people who are knowledgeable in a specific subject in the organization. As a team leader at UOL stated, "*I have a problem to solve here in my project*", then '*Oh, I do not know how to solve it, but there is someone who knows*'".

The Purpose of Knowledge Levelling This purpose is mainly referred to promote knowledge redundancy, technical excellence and greater understanding of the organization's and project's processes. As an interviewee at Apontador said: "*We encourage dissemination of knowledge through, for instance, leveling testing culture from a project to another one.*"

The Purpose of Providing Feedback/Visibility This purpose is crucial for leveraging improvements and awareness to other teams. As a developer at Caelum stated, “It is important to gather feedback from other projects and not be biased by the team approaches to the problems.” A department director at UOL sums up, “Without promoting visibility to others, it becomes more difficult to learn aspects that are more distant.”

The Purpose of Problem Solving This purpose involves taking joint actions to solve problems, to share knowledge in practice, and rethink or restructure their assumptions. As a ‘Scrum Master of several teams at UOL posited, “The need to solve a problem drives to a lot of knowledge exchange”.

The Purpose of Improving or Reusing Solutions Considered to avoid “reinventing the wheel” and benefit from existent solution. A team leader at UOL stated, “I think the main focus is to reuse solutions, or sometimes, implement the most appropriate solution to the problem you have with an existing one, more robust, and best addressing the problem.”

The Purpose of Improving Workspace Environment According to our informants’ point of view, KS also improves professionals’ interactions and strengthens their relationships.

The Purpose of Developing Insights or Innovating KS is considered as a promoter of insights and innovation. Caelum’s development manager stated, “We need to study new things (..) we have to be ahead”.

Figure 2 sums up the practices by purposes, while Table 8 lists the work practices (and its variations) employed by the different organizations. Each practice is labeled by the level of purpose achievement as totally meets (dark-grey) and indirectly meets (light-grey). The rest of this section will describe each one of these practices.

Practices	Purposes						
	To id. k/ledge. owner	To level k/ledge.	To prov. feedback/visibility	To solve probs.	To improve solns.	To improve w/space.	To dev. insights/innovate
Face-to-face conversations at workspace							
Informative workspace							
Rotation among teams and projects							
Collective meetings							
PP between different teams							
Technical presentations							
Marathons							
Coding Dojos							

Totally meets
 Partially meets

Fig. 2 Practices by purposes

Table 8 Practices by company

Practices		Organizations	
UOL		Apontador	Caelum
TW-Brazil			
Face-to-face conversations at workspace environment	Are hindered by a workspace with fixed workstations and walls of half heights. They provide shared spaces with food and drinks to foster inter-team interaction	Occur across teams of the same product in a semiopen workspace	Occur freely across all development teams in open workspace
Informative workspaces	Interrelated teams place their informative workspaces close to the same partitions	Interrelated teams place their informative workspaces side by side	Team members have total access to other informative workspaces
Rotation among teams and projects	When composing new or existing teams (to overcome challenges)	Freely employed by interrelated teams	Only when composing new teams
Collective meetings	Stand-up meetings with all sub-teams from the same product biweekly, overall planning after sub-teams' planning meetings, review meeting with other teams' representatives, and sparse general retrospectives	Planning, review, and retrospectives meetings with three sub-teams	Daily standup meetings with all members and general retrospectives periodically
PP between different teams	Only adopted when necessary	Fairly employed by the 3 sub-teams	Applied only within teams

Table 8 (continued)

Practices	Organizations	Apontador	Caelum	TW-Brazil
Technical presentations	UOL	Occasionally promote internal seminars	Technical lunch weekly, “TechDay” every 6 months	Demos for showing different designs for others, occasionally employ ‘Lunch and learn’
Marathons	They occasionally promote internal seminars and present a developed complex technical solution to more experienced people N/A	N/A	Technical competitions during lunchtime, “Programming Sunday”, and 48 h of competition between projects to open source community	N/A
Coding Dojos	N/A	Employed biweekly after working time with general challenges	Employed occasionally during lunchtime for the development department with specific challenges from their ordinary projects	Mostly employed in external community events

5.1 Face-to-face Conversations in the Workspace Environment

Most teams work close to each other. Due to proximity and organizational structure, face-to-face conversations are facilitated in the context of the studied organizations. UOL and Caelum are exceptions because there are walls in the offices that hinder people to talk freely and ask questions in the workspace (even walls that are of half heights only).

A team leader at UOL depicted a recurring situation, “*Suppose I use a specific technology. Then, on the side, and it is really aside, there is a guy who is racking his brain to do the same thing*”. They have mitigated this problem by providing spaces close to workstations with food to attract spontaneous and informal conversations. Similarly, at Apontador, they have created shared workspaces like coffee and gaming rooms.

On the contrary, Caelum and TW-Brazil have plain tables and seats without arms to facilitate Pair Programming. As we mentioned in Section 3, both companies are experienced in ASD and assign spontaneous and informal interactions to their settings: “*Open workspace spreads out information of all kinds without barriers, and many people with different roles sit nearby. If you have a conversation across the table and another one knows what it is, (s)he will enter the conversation, hear, become aware (...) in a few moments, everyone already knows more or less what your project is doing, what you have just done*”.

Open workspaces help in identifying the knowledge owners for knowledge transfer, as stated by an interviewee at UOL: “*This guy knows that system very much and this one doesn’t, so let’s put them together. So we use a little of the knowledge matrix*”.

Another great use of open workspaces is spatial rotation. Teams usually change their position in the room. A team leader at TW-Brazil depicted, “*If my team is aside with another team for a long time, they start to have vicious cycle of having the same conversations, sharing the same knowledge. We switch teams to different places. So, this helps the team not to create a routine, and exchange other opinions, arguments, and discussions*”.

As mentioned before, workspace environment provide great value, but software developers have to deal with the problem of noise. Sometimes they have trouble in moderating people, as the development manager at Caelum reported, “*Some people prefer partitions . . . soon we will create a space with glass walls for those who want to have a more isolated and silent space*”.

In general, this practice relies on openness and freedom of communication, company size, learning-oriented strategy, and culture.

5.2 Informative Workspaces

The studied organizations implement informative workspaces and virtual tools for managing their project tasks. At Apontador, interrelated teams place their informative workspaces side by side, as a developer said, “*Our posters and boards are also seen by other teams. I think this improves our project visibility to others*”. At UOL, they place interrelated teams in common environments.

Caelum and TW-Brazil team members, who are located in open workspaces, have total access to other informative workspaces, so they become more aware of what is happening in other projects - even if they are not involved. They can even ask question in these informative workspaces. At Caelum, people usually see other informative workspaces, become aware of what is going on in different projects and question other developers from these projects. They also promote common informative workspaces for the area where members share problems to be solved by others. Finally, they also share their teams’ week goals in a common board.

At TW-Brazil, awareness among team members occurs naturally because of the spatial rotation and the movable white-boards. They say that people are curious and critical to evaluate and learn from cross-teams informative workspaces. They usually use eccentric ideas to catch the colleagues' attention and foster interesting discussions.

To sum up, in order to reach other teams, informative workspaces rely on overall commitment, willingness to enhance these workspaces, and collaborative behavior.

5.3 Rotation Among Teams and Projects

Team member rotation is undertaken by the organizations to compose a new team or to integrate a new member to an existing project. A member at UOL said, "*We have done that a lot, kind of on purpose (. . .) we take a person from one team and place her/him in another one to encourage knowledge exchange*". Another interviewee from UOL added, "*We say that even if you are in a team now, it does not mean that you will be part of this team forever. If needed, we will move people across teams. So people end up knowing other parts of the application*". At this company, rotation is promoted when a team has problems in specific roles that were already overcome by another one. Rotation occurs either within the sub-teams of the same product and other teams. For instance, the culture of doing tests, which is critical for them implies that everyone in the team has to create tests, so this knowledge generally needs to be disseminated. Therefore, team leaders agree to rotate members to help with the dissemination of the testing culture and when it should be performed. Team members return to their original team once this knowledge is disseminated.

At Apontador, we observed that they freely employ rotation of members of interrelated teams, especially to solve problems or to level knowledge about a specific work process, technique or technology, e.g. TDD. At Caelum and TW-Brazil, they usually rotate team members when composing new teams.

Regardless of the company size or experience on agile methods, they all consider this practice crucial to share knowledge, because of its usefulness and lower cost. It relies on leadership alignment to bridge the gaps, developers' willingness to KS, and organizational vision and alignment.

5.4 Collective Meetings

Collective meetings are generally used by interrelated teams. At UOL, they have also adapted their Scrum of Scrums meetings into other ceremonies that involve different teams from the same product. They undertake stand-up meetings (Beck and Andres 2004) with all sub-teams from the same product biweekly (known as "Mega Daily") and overall planning ("Mega Planning") after sub-teams' planning meetings. Besides, these ceremonies make people aware of the stories and raise questions or interdependencies. The impacts are discussed and when necessary, they schedule other meetings to detail specific issues about related activities. And finally, they have a review meeting to present what was developed in the iteration and one person of each related team is designated to attend these meetings.

They used to adopt study groups until the Scrum implementation. As a member outlined, "*At the beginning, I thought it was very important to attend, because we have exchanged knowledge among Scrum Masters, POs, etc., but it lasted only for a short time, just until reaching the focus of addressing the original need, which was the Scrum implementation*".

At Apontador, also due to interrelated products, they employ Scrum meetings jointly, such as planning, review, and retrospectives with three sub-teams. At TW-Brazil, in teams from the same customer, they also make daily standup meetings with all the members: "As

our team is getting too large, standup meeting does not fit into fifteen minutes. So, the sub-teams make their own standup meeting quickly, just before the meeting with other teams. Then, only one person speaks for the team. It is more focused (...) people put forward challenges and new things that they have learned”.

Agile retrospectives (Derby and Larsen 2006) involving several teams encompass sub-teams (about the same product/customer), departmental, or company retrospectives. Within sub-teams, UOL, Apontador, and TW-Brazil schedule retrospectives periodically. At TW-Brazil, teams from the same customer have established “walling walls” where teams’ members stick notes about overall issues. Prior to this meeting, the notes are selected in order to guide their inter-team retrospectives.

The other variations of retrospectives generally do not follow a schedule. Caelum and TW-Brazil employ departmental and company retrospectives in order to reflect on what to improve in the organization. They use boards to register topics for each retrospective.

Companies that establish a clear organizational strategy toward continuous improvement, integration among projects and teams, and willingness to bring all people together are more likely to scale retrospectives to the organizational level. Most interviewees stressed the need to have a moderator, assign actions to responsible people and monitor the chosen actions.

An interviewee from TW-Brazil stated that extensive documentation in wikis does not work at all for them. That person stated that documents are not as rich as the experience developers have in projects, as they confirmed, “*Knowledge is collective and the context of the software is mostly on developer’s head (...) that is why you have standup meetings, demos for showing different designs for others, as a way to try to spread this context, since it is much more than what someone wrote in the document (...). Documents end up as dead reference.*” So, they only document what really is necessary, in order to avoid overhead and to write good and readable code with automated tests.

5.5 Pair Programming Among Different Teams’ Members

The studied organizations engage PP among different teams’ members occasionally and specifically for leveling knowledge (mentoring); for associating with rotation to handle on an isolated action; and for scaling knowledge.

At UOL, PP is only adopted when necessary, for instance to overcome a specific role or to solve a problem. At Apontador, it is fairly often employed by the interrelated teams.

At Caelum, we observed that PP is part of their software development culture, so they freely apply it with as many people as possible. In their point of view, the broad rotation of pairs improves knowledge propagation, fosters visibility to other projects, and enhances collective code ownership and internal software quality. Lately, they tried to foster cross-team PP by using a matrix to reward people pairing with at least 70 % of the developers, but after 3 months of no one winning, they realized they did not have availability to make it. So, they abandoned this metric and just continue to employ the cross-team PP freely.

At TW-Brazil, PP is also part of their culture within teams. They work in pairs seven hours a day in most working tasks within the team (local) and with distant teams (remote), such as PP, writing e-mails, preparing internal or external presentations, doing book translation and even in tasks that are not work related. A member stated, “*Pair programming is a rule, you rarely see someone working alone here (...) At the end of the day we are exhausted, but we know that we worked intensively*”. However, switching pairs is limited to teams of the same customer, due to customer requirement. PP is employed for leveling knowledge to novices, as a leader expressed “*We do not believe in learning about a system by*

reading a stack of documents. We do pair programming with novices, so learning becomes faster”.

Cross-team PP depends mostly on the software development culture, on the willingness to explain and share knowledge, and the integration between projects and teams. At UOL, resistance is outlined by a director, *“Here we do not force adoption of any agile practice, so developers chose not to adopt PP because of the infrastructure. They also feel uncomfortable to pairing with some people”.*

5.6 Technical Presentations

Informal technical presentations are promoted without a pre-defined agenda and in an ad-hoc manner. At UOL and Apontador, they occasionally promote internal seminars to disseminate knowledge about specific topics and knowledge apprehended in conferences attended by team members. An interviewee claimed that these internal seminars continue to work because *“It is more than an exchange of knowledge, it is an accountability”.*

At UOL, when they have to develop a complex technical solution, they usually present it to more experienced people from other teams to evaluate and/or improve it. Once the solution is quite mature, they also present it to everyone in the area. At TW-Brazil, they also undertake *“demos for showing different designs for others”.*

Formal presentations about technical issues are promoted in pre-defined agenda and require the participation of most of the teams. At Caelum and TW-Brazil, they organize technical lunches (also termed as ‘Lunch and learn’), these are presentations during lunchtime, which is offered occasionally by the organization. At the end of the session, they make a retrospective to sum up collective learning and raise feedback for improvements.

At Caelum, technical lunch used to be held biweekly, but after a few months, they decided to have it every week. Knowledge is considered a competitive advantage in their business domain. As the development manager said, *“The study has become a natural work practice (. . .) it is clearly 30 % of the work time”.*

They also employ “TechDay” every six months, as lectures on current technological innovations. This practice was also stimulated by UOL, but did not work too long. As a member depicted, *“Sometimes, it is really cool to bring up new things, but sometimes we are so overloaded with work, then you have to do a presentation for two weeks on a new technology. So, it ends up just not working”.* Another member said, *“Once I started the presentation and did not finish it, but as nobody charged me, I did not continue.”*

Another approach that has failed at UOL, was a presentation involving software development and business domain teams. A Scrum Master reported, *“It lasted three meetings”.*

In coaching sessions, the specialist also needed to adapt some practices to improve learning, *“I have used open space-based sessions⁴ with very positive results. BoF sessions leave things in the clouds, they are very good to share, but not to generate results”.*

Successful initiatives depend mostly on organizational commitment to consider KS practices as part of their work processes. As a consequence, organizational commitment reinforces the KS behavior and the KS culture.

⁴Open space session is based on the open space technology (Dierkes et al. 2003), an elaborate facilitation technique usually held on conferences (<http://www.agileopen.net/on-open-space>), to provide really productive meetings.

5.7 Marathons

Companies focused on creating innovative products/services employ or have employed marathons to foster a discipline and a culture of continuous learning and KS behavior among professionals.

The head of the R&D department at Apontador, said that they encouraged the “Marathon of Innovation”, by releasing the company space, infrastructure, and technological frameworks for employees to develop their own ideas in two weekends. Participants composed multi-functional teams (e.g., POs, Scrum Masters, and developers) that were different from their routine teams in order to foster the mix of people with common interests. The top three best projects were awarded and became open source contributions, so the winners would take time for maintaining and continuing the projects. He stated that the participants really enjoyed this practice as it promoted learning and the creation of innovative products, and they were keen to repeat it further. However, by the time of that marathon, the company started to receive several investments from new stockholders. The company had a sudden growth of employees and projects, which changed some of their priorities, and then they eventually discontinued the projects. As he reported, “(. . .) *people got frustrated after noticing they would not be supported anymore to continue the projects.*”

At Caelum, they undertake several marathons because of their business domain. In the weekly technical lunch, they form teams to exercise programming: “*We set two teams to refactor the same code*”. Another marathon is the “Programming Sunday”, which happens every two or three months, where pairs are formed to code several chosen internal or open source projects. The company offers lunch. In a recent adaptation of this endeavor, they are employing 48 h of competition between projects.

5.8 Coding Dojos

This practice is only employed by Apontador and Caelum. They usually adopt it to improve programming skills (Sato et al. 2008).

At Apontador, after participating in Coding Dojos of the agile methods community in São Paulo, they noticed that this practice would help them improve integration with other areas and support learning in the workplace. They started to employ Coding Dojos by inviting all the teams in the organization through their communication channels (mailing lists, intranet, face-to-face invitations, etc.).

This practice was promoted to support learning in the diverse programming languages used in the organization. It was held biweekly after work, so it would not affect liberation of employees and would not harm the daily routine of the areas. They decide the programming language to be trained and one knowledgeable person is responsible to define the challenge. At the end of the session, they conduct a retrospective to get feedback from the participants and define improvement actions and topic for the next session.

We observed that the majority of the audience appreciated the practice and in the retrospectives (conducted at the end of this practice), there were several statements praising the initiative. As a member stated, “*With the Dojos different teams are learning about automated tests and programming best practices, which seemed to be impossible in the past*”. Another participant declared, “*It also creates a sense of unity in which all realize that we work in the same company, with common objectives and that the company is more than just ‘our own team’*”.

However, as it takes place after working hours, sometimes the number of participants is low and due to delivery pressure, they often need to reschedule these events. As a member

declared, “*It is difficult to keep discipline and do the sessions even with only few participants. For about one month ago we had two sessions. I believe that now we are more aware of that.*”.

At Caelum, Coding Dojos are also employed outside working hours. In this company, as they are more committed to enhance learning because of their business domain, they are more disciplined in conducting the sessions that are usually held in technical lunch sessions. In the beginning, they were selecting all kinds of challenges not particularly related to the work and were focused on just doing the sessions, as a member said, “*In the past, Dojo for us was just fun, now we try to add more value by choosing challenges from the daily work*”. They realized that it becomes much more effective when they establish more focused sessions.

In our observations, we noticed that developers and leaders are sticking notes on a poster for the next session (as we declared before, they also have boards and posters for general purposes). The notes were about their suggestions of challenges from their daily work (e.g., recurring project problems) or for further project endeavors to be considered for a Coding Dojo. Others occasionally read the notes and vote for the challenges that they consider more interesting and relevant.

At TW-Brazil, Coding Dojos are mostly employed in external events, e.g. software development community workshops, agile methods events, etc.

Coding Dojos are considered for purposes of leveling knowledge, solving problems, improving existing solutions or developing new ideas in technologies.

The mentioned agile software development organizations deal with inter-team knowledge sharing by applying practices to foster social interactions. Even being experienced companies in agile software development, each company apply these practices on their own way. We observed that the perceived differences are related to influencing factors that affect their way of adoption and consequently their effectiveness. In the next section, we describe these influencing factors and the perception of effectiveness according to the participants.

6 Inter-team Knowledge Sharing Influencing Factors and Effectiveness

As referred before, we identified differences on how the studied organizations cope with KS and achieve its effectiveness. These differences encompass influencing factors that empower or hinder inter-team knowledge sharing. We found that influencing factors are composed by organizational conditions and stimuli.

The more knowledge-driven are the organizational conditions and stimuli, the more the practices for inter-team KS are created, sustained, adapted, and even abandoned spontaneously, i.e., the influencing factors determine the achievement of inter-team knowledge sharing effectiveness for the company.

Before the selective coding process, we analyzed the similarities and differences of these influencing factors across the studied companies. Both are presented in the Sections 6.1 (Organizational Conditions) and 6.2 (Stimuli). Further we present evidences regarding the inter-team KS effectiveness, and lastly, we gather our findings in a conceptual model for inter-team KS.

6.1 Organizational Conditions

The organizational conditions identified in this study are strategy, structure, culture and individual behavior, environment, top management and leadership support, communication flow and channels, integration among teams and projects, and agile method(s) adoption.

6.1.1 Strategy

Regarding strategy, the studied companies are very concerned about achieving fast delivery of value to customer. On the other hand, we observed differences on how they cope with strategic alignment and knowledge sharing as part of their work processes.

Some companies are very concerned about applying KS effectively without affecting their agility in delivering software. These organizations assign their difficulties in sharing knowledge across teams to pressures of time, customer, and/or business domain that require quick and frequent delivery of working software. As a scrum master at UOL declared, *“I think the pressure for delivery yields a lack of time, because a bias of Scrum is the great focus on delivery, delivery, delivery. Then, practices like the ones I am trying to adopt with the team, more behavioral practices get lost [the scrum master was talking about a group practice to integrate the team]. Then when we get to the end of the year, some teams negotiate to stop the sprints to ‘clean up the house.’”* This difficulty is also stated by a director at UOL, *“Exchange good practices is a challenge, since our major focus is on the product (. . .) You forget the other teams that work with different products.”*

Likewise, at Apontador, in the Vision statement, there is an explicit aim to share knowledge and experiences for the company innovation and cohesiveness. Also, there was a poster in the software development department about the organizational perspectives for the near future. The main aim was to consolidate the company as reference in their business domain. There was also an aim that the organizational knowledge would be preserved for continuous growth. However, the R&D director stated, *“The company is undergoing a growth stage, accompanied by several changes”*, and the scrum master added, *“Our team feels that is under constant pressure for delivery in the end of the sprint”*. In the teams’ retrospectives at this company, some topics were presented more frequently, such as the pressure for delivery in a one-week sprint would impact the delivery of value to customer with technical debt, and low quality and innovation. Other team members emphasized the need to improve the communication, and the need to create a momentum for discussions to reduce rework.

On the contrary, the specialist declared, *“Time constraints and focus on delivering value are not justifications for the lack of knowledge sharing, but a means to build such justifications”*. Other organizations create an organizational posture toward knowledge and employ more practices for KS, which are more frequent and become naturally part of their work processes. At Caelum, learning is a fundamental aspect of their daily routine, as the software development manager explains, *“Learning has become a natural practice and it is not expected someone to stay for a long time in the same project, but enough time to get mature in its technologies, design, architecture, domain, and to help improving it.”* Recently they started to promote technical lunch sessions weekly. Yet, they have abandoned the PP matrix and started to adopt other approaches, such as PP with external teams, projects marathon and goals board. They also replaced the daily meeting with everyone for a weekly meeting.

An important aspect raised by the specialist is that *“The organization must establish organizational level commitment, shared vision, responsibility, sustainable pace, and motivation to achieve a balance in shipping software and accomplishing long-term strategies.”* It is undertaken, as he reported, *“By calibrating the team. This means carrying out activities through which the team gets mature in terms of project vision and also making team members understand that the organizational gears, which are sometimes unfamiliar to the project’s reality, are essential to the continuity of the organization”*.

6.1.2 Structure

We observed that the physical and hierarchical structure of the agile companies have provided more teams' closeness (team members sit nearby). However, this influencing factor has also assigned differences that affect the level of hierarchical formalities, physical barriers, autonomy and self-organization.

At UOL, the hierarchical structure still has several levels, but they have changed the composition of teams. Now, they have multi-functional teams, as a scrum master stated, "*Some departments disappeared, other functional departments still exist, but with a different characteristic than before. Other departments were put together to become part of the teams, such as quality assurance, webdesign, database administration and deployment.*" At Apontador, directors and top management work in their own spaces and do not share the agile teams physical workspace. The hierarchical structure is organized in three levels: directors, management and operational.

The office layouts of UOL and Apontador are composed by big rooms with fixed workstations and furnishings for each professional. As a member of the top management at UOL stated, "*People are in the same workspace and sit nearby, they know each other much more (...) but still it is not the physical structure that we think is appropriate. We saw a company that is structured for XP, the office was set up in order to meet the process and it is quite different (...) the layout has to be adapted so that two people can work in the same machine.*" Just by having a proper structure, the process becomes more facilitated.

At Caelum and TW-Brazil, the workspaces are characterized by plain tables without partitions, and chairs without arms. The tables are placed in the middle of the office, so the walls are used for informative workspaces. There are fewer hierarchical levels and people from different levels sit next to each other. Professionals work with laptops (not desktop computers) to facilitate their mobility and freedom to sit wherever they want in the workplace. These workspaces also facilitate the adoption of pair programming. There are also couches, tables and chairs for sitting around informally, and a projector area for presentations and virtual video-conferences with distributed teams. As the software development manager at Caelum defined, "*Our room is an open workspace. There are tables with no partitions and no walls. People can sit wherever they want, next to anyone else. The developers themselves chose to have this freedom instead of their own tables or seat. (...) As all members are together, this encourages people to make questions anytime. Anyone can answer and sometimes a debate will arise.*"

6.1.3 Culture and Individual Behavior

Agile software development teams achieve great levels of commitment, transparency and responsibility, as expressed by the scrum master at UOL, "*The developers are much more committed to the product.*" However the organizational culture ends up operating as boundaries to values and attitudes toward knowledge, such as solidarity, mutual trust, freedom, tolerance for admitting mistakes, and shared understandings to legitimate work processes. As explained by a team leader at UOL, "*Within the team, this [KS culture] can even exist, but across teams, forget it.*" And another colleague added, "*In general, what goes along with it is the word 'problem'. (...) If we were not talking to each other because of the need, they would be working there and we were working here separately.*"

At Apontador, the scrum master explained, "*When our teams [sub-teams of the same product] identify problems in other teams [sub-teams] or when we find out a good solution*

to our team [sub-team], it is common to think of taking this knowledge to other teams [sub-teams], but we hardly think of disseminating knowledge to teams of other products of the company.” Because of their difficulty in interacting with other teams of the company, the scrum master started to adopt *Coding Dojos* by inviting people from the whole company, as the scrum master stated, “*This [practice] also creates a sense of unity, in which everyone realize that work in the same company, with common goals and that our company is bigger than our ‘own team’.*”

On the contrary, at Caelum we could observe this organizational cultural shift, as a developer explained, “*Here we encourage several experiences and any other social activities amongst developers help in creating a stronger department feeling where everyone feels more comfortable talking to each other. This freedom helps developers feel free on being critic during pair programming, not to feel shy while showing code they are not quite proud of, or ask maybe foolish questions to the team.*” The developers have the freedom to actively manage their own time, studies, and growth within the company. Instead, they are expected to execute a high quality job with excellent feedback from the customers. This freedom requires, as described by the software development manager, “*a level of maturity from the developers*”.

Likewise, at TW-Brazil, as they have open workspaces freeing barriers to share knowledge across teams, people feel free to share knowledge, however as they have strict customer constraints, their interactions across teams are not totally free, but more carefully established.

To change organizational values and principles, it is necessary a complex endeavor, as the specialist said, “*To change values is too complicated (...) few trainers or coaches know how to make this change (...) trying to reach the root of the problem.*” By his experience, top management is usually resistant to embrace wide changes.

6.1.4 Environment

We noticed that the organizations had to deal with environment tension of change and stability, so learning could take place. The creation of a momentum to learning was related to the business domain, but relied much more on the organizational strategy and culture toward knowledge. The development manager at Apontador reported, “*The pressure for delivery was high and we did not have the discipline to continue [He was talking about doing dojos more often].*” In a *Coding Dojo* retrospective, it was raised as a drawback the small audience because other team members were working on a task force.

At UOL, a team leader stated, “*The ideas come up, are supported, but their implementations are not taken seriously.*” Another scrum master at this company added, “*This happens because actually our focus is on the customer needs. I am not going to be searching something to learn if I have a problem that will affect my team goal.*”

Caelum’s environment is quite different, because of their business domain, the creation of a momentum to learning and teaching is fundamental. So, they exercise both aspects as part of their work processes. As the software development manager explained, “*The instructor’s goal is not to make the student memorize APIs (Application Program Interface), rules, patterns or practices, but to be able to understand the problems and construct the solutions on their own. All these ideas matter because, besides teaching outside students, employees teach each other during their software development activities. They quickly realize the value of that freedom and the responsibility that it takes, usually adapting themselves fast to such environment.*”

6.1.5 Top management and leadership characteristics

Top management and leadership characteristics are also relevant, as a Scrum Master at UOL stated, “*Top management is more centralized and does not give much freedom to experiment, but leadership is less imposition and more transparency. These aspects are somehow related to Scrum, which makes people decide (. . .) but of course, it needs to have leaders that give these conditions.*”, and also a team leader reported, “*So we support, encourage [KS], but when it’s time to put it into practice, we just prioritize other things*”.

At Apontador, these characteristics are similar, for instance, a scrum master depicted, “*I have some concern of how it [Lighting talks] will be seen by the directors.*” This scrum master fosters *Coding Dojos* at the company, however the directors seem not to be aware of the importance of this initiative, as expressed in the following statement, “*The room reserved for the Dojo was in use by one of the directors. We had to switch to another room. The Dojo members pointed out in the retrospective it was an aspect that did not work well and should be improved, but we did not state any action taking. I think the directors seem not to see value in this.*”

Otherwise, at Caelum, the high support for improving expertise and delivering high quality services and products comes firstly from the founders. The employees have freedom to set up their working time and to contribute to open source projects. They have established one day per month to stop work and refactor code or contribute to open source projects, along with several other practices.

6.1.6 Communication Flow and Channels

The communication flow and channels are valued by the studied organizations. They are concerned with improving this influencing factor as a way to improve KS through the use of several tools, such as intranet, wiki, repositories, etc. However, the differences come to the level of osmotic communication established by the companies, which relies on the openness to communication and the perception of its benefits by improving interactions and using tools to help in raising problems or sharing topics of interest. The specialist claims that ASD impact on communication flow and KM processes because “*agile methods promote interaction of various forms and the team is involved in everything*”.

At Caelum, due to mutual interplay of physical structure and culture, communication flow is highly facilitated, empowered and it occurs without obstacles. We observed the use of several communication channels, e.g., mailing lists, forums, or even informative workspaces to share sketches. For instance, an internal discussion list allows anyone to ask questions and share any idea, even when not physically present. All interviewees mentioned that the level of the discussions that happen in there is high.

Otherwise, at UOL and Apontador, they pointed out the use of similar channels for communication, however, there the information soon becomes outdated, as the director of R&D at Apontador depicted, “*People started to use only in the beginning, when it was considered a novelty. After some weeks, the flow [of information] nearly disappeared.*”

6.1.7 Integration Among Teams and Projects

Integration among teams and projects is another relevant factor. The organizations generally consider inter-team KS when the teams have areas of overlap and integration needs. As we could observe in several previous statements, most interaction is established when teams have common purposes.

An interesting aspect involving this influencing factor is that we also observed that organizations with great openness to communication and cohesiveness create integration across teams, even without strict needs for that. At Caelum, the software development manager assigns this behavior to their clear organizational purpose, as he explains, “*It is clear for everyone in the company that they are there to assure the excellence of our purpose as a company, not only as an individual or as a team.*”

6.1.8 Agile Method(s) Adoption

The agile adoption that encompasses the number of agile methods adopted, the level of adoption in the organization (e.g., team, department, organization, etc.), the time of implementation and the level of maturity (beginner, intermediate, advanced) also impacted on the organizations as they became more prone to embrace changes and learn.

Even being experienced companies on agile methods, we observed differences in the way the companies cope with agile methods in an organizational level. At UOL and Apontador we found several difficulties in scaling agility that seem to be similar in scaling KS across teams. On the other hand, at Caelum and TW-Brazil, as being agile companies since their foundation, we could identify differences on how they deal with organizational agility and KS at the organizational level.

An intricate relationship of strategy, structure, culture, environment and top management support was identified. Strategic directions and environment impact on culture, structure, and top management/leadership support. Culture conducive to learning is related to physical and hierarchical structure, to strategic directions, to environment and also to agile method(s) adopted. Structure and also agile method(s) adopted impact on communication flow. Integration among teams and projects is also related to strategy, structure, culture and environment. Table 9 sums up the conditions by company. Because of lack of space, **Size** was one of the influencing factors that we did not scrutinize because it is considered a common difficulty in scaling agility and likewise affected inter-team KS effectiveness.

6.2 Stimuli

The interaction of people across teams is triggered by different stimuli. The main motivators for KS were identified as:

- **Problems**, as a member at UOL reported, “*It is related to the project, a problem*”.
- **Common goals/interests**, as a team leader at UOL added, “*It’s the need; we help each other because of a common goal*”.
- **Incentives**, companies stimulate their professionals by offering incentives, such as lunch to engage in technical sessions, marathons, and Coding Dojos.

The specialist stated that beyond all stimuli, what really motivates people to share knowledge is “*The feeling of responsibility for the development of the product (or service)*”. We realized that most organizations focused on short-term accomplishments and poor organizational posture toward knowledge, are stimulated by problems, common goals and incentives in reactive inter-team KS strategies. Otherwise, organizations focused on long-term strategies and posture toward knowledge, apply more proactive strategies.

The aspect of **sustainable pace** is something still under discussion in the organizations. According to an UOL respondent, they often have established a “*very maddening pace of delivery, delivery, delivery*”, as time is not fully considered as a resource for learning and sharing knowledge. Others try to accomplish practices after work not to affect liberation

Table 9 Summary of conditions by company

Conditions	Organizations		
	UOL	Apontador	Caelum
Strategy	Short-term focus on delivery of working software	Short-term focus on delivery of working software	Consistent, long-term focus
Culture and individual behavior	People feel uncomfortable to expose what they do not know	Transparency, collaboration, respect, voluntarism and informism	Transparency, collaboration, respect, and informism
Structure	Fixed workstations with walls of half heights, several hierarchical levels, people have difficulty in accessing others	Semi open workspace, few hierarchical levels, people of the same department are accessible	Open workspace, few hierarchical levels, all people accessible
Environment	Have difficulty in questioning their assumptions	Have difficulty in questioning their assumptions	Reconsideration of assumptions
Size	Large-sized	Medium-sized	Medium-sized (branch office in Brazil)
Top management and leadership support	Medium (High leadership support, but few top management support)	Medium (High leadership support, but few top management support)	High
Communication flow and channels	Medium and use of few communication channels	Medium and use of few communication channels	High support and use of several communication channels
Integration among teams and projects	High only within teams of the same product	High only within teams of the same department	High only within teams of the same customer
Agile adoption	Fairly mature on Scrum (adapted with some XP practices), which is adopted for 4 years in one department of the company	Fairly mature on Scrum (adapted with several XP practices), which is adopted for 3 years in one department of the company	Mature on adaptation of XP, Scrum and Kanban, which is adopted for more than 10 years by the whole company

Table 10 Summary of stimuli by company

Stimuli	Organizations			
	UOL	Apontador	Caelum	TW-Brazil
Problems	x	x	x	x
Common goals/interests	x	x	x	x
Incentives	x	x	x	x
Sustainable pace			x	x

of employees and harm the daily routine of the departments. An interviewee at Apontador said, “*We have difficulty in defining a sustainable pace in which knowledge sharing is part of the activities during the sprint*”. Table 10 lists the stimuli by company.

6.3 Inter-team Knowledge Sharing Effectiveness

According to the participants, the inter-team knowledge sharing effectiveness is evaluated by four components:

- **Level of purpose achievement:** Each KS practice can address one or several purpose(s). This component evaluates the level of purpose achievement: (1) does not meet; (2) partially meets; and (3) fully meets. The higher the level, the more the practice is likely to be effective for the respective purpose.
- **Frequency of adoption:** We identified the following frequencies: annual, semiannual, quarterly, bimonthly, monthly, occasionally (when there is no rule to happen), biweekly, weekly, daily, and freely (when there is no rule to happen, but it has to happen). This component subjectively depends on the practice and organizational context to be considered effective.
- **Level of formalization:** As the agile culture nurtures an environment of low imposing, organizations tend not to force the institutionalization of the practices, in order to foster a natural adoption. As a member of top management at UOL reported, “*Learning has to have a low cost. To have a low cost, it has to be natural*”. We identified the five levels of formalization: the practice is not adopted anymore; the practice is still little known and informally adopted; the practice is widespread and accepted, intended to formally adoption; the practice is widely accepted and formally adopted; and there is no intention to formalize the practice, since it should become a natural adoption; and the practice is naturally adopted, which means widely acknowledged by the organization. The higher the level of formalization, the more the practice become part of their work processes.
- **Reassessment:** As the KS process is dynamic, the participants reported the need to periodically make reassessments on the practices and the conditions to evaluate if they are still being effective in their context. As a member at Caelum stated, “*We reassess if the practice continues to bring value to the department, if people are participating or collaborating in interactions or if they are isolated*”.

6.4 Conceptual Model for Inter-team Knowledge Sharing

The last grounded theory coding step is the selective coding. In this step, we achieved an emergent conceptual model that is illustrated in Fig. 3. This conceptual model is based on the studied companies and is our main contribution after a thorough grounded theory

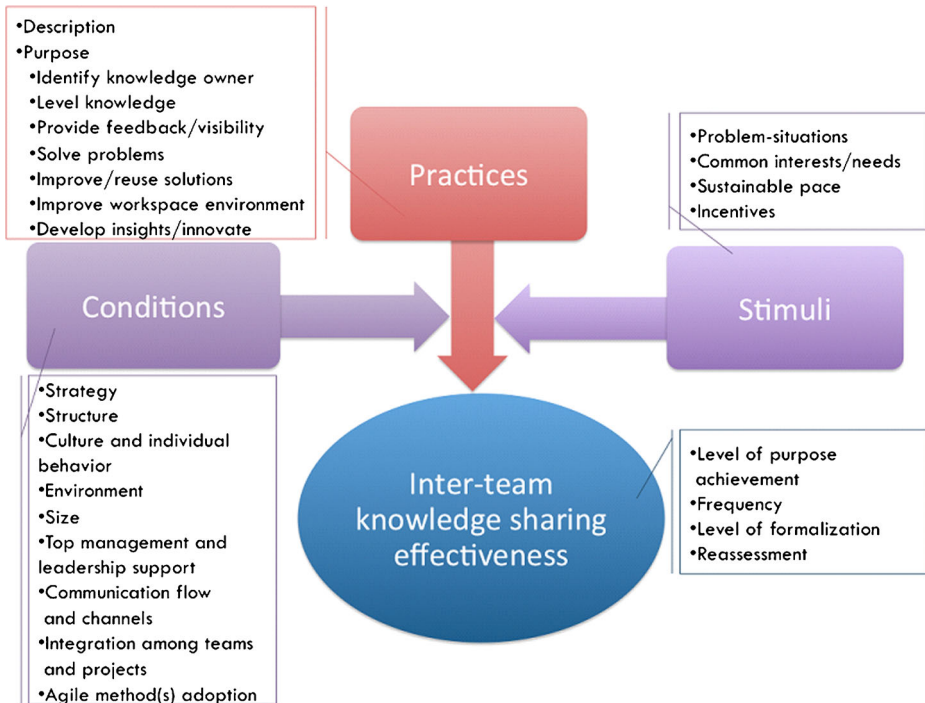


Fig. 3 Theoretical model for inter-team knowledge sharing effectiveness in ASD

study. This model explains that **inter-team knowledge sharing effectiveness** in agile software development organizations may be achieved mostly by the application of **practices for socializing knowledge**. The effectiveness of these practices is affected by the **organizational conditions** and **stimuli**, which are influencing factors that empower or hinder the knowledge sharing process. The practices have a **description** and a classification according to specific **purposes**. Effectiveness is evaluated by the **level of purpose achievement, frequency, level of formalization and reassessment** of the practices in the organization.

In Fig. 4, we show evidences of the relationships of the components through an instance of the conceptual model to a practice adopted by one of the organizations. Due to the lack of space, we selected only one instance, but during the data analysis process, we found out that this relationship is present in all of the observed organizations. We explain this instance below.

First of all, this company considers knowledge as competitive advantage in their business domain. There is great support to improve expertise and deliver high quality products and services, as the development manager explained, “*The support for sharing knowledge here is total, because we need to be knowing the subjects [software development topics] (...) This support is critical because of the company’s role. The company could not live without encouraging this. (...) We’re ahead, so we can teach them*”.

For that reason, their strategy, structure, culture and individual behavior, environment, top management and leadership support, and communication flow and channels are positively impacted by their clear commitment to enhance the overall knowledge. As well, we observed that their agile adoption and size contribute to such conditions. Their commitment

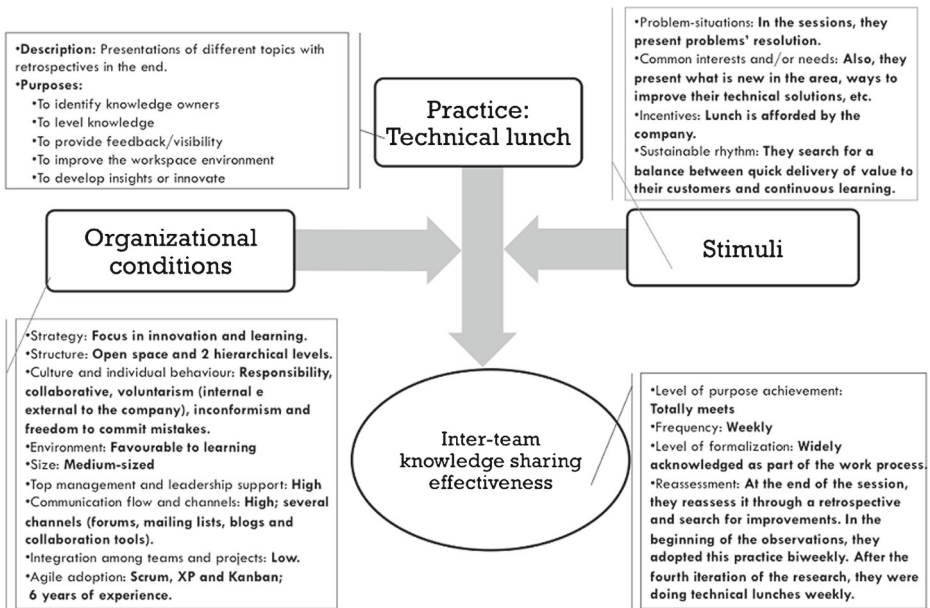


Fig. 4 Instance of the theoretical model for technical lunch adopted by Caelum

toward knowledge positively impacts all their stimuli to share knowledge. For instance, they establish a sustainable pace, as the development manager says, “*The study has become a natural work practice (. . .) it is clearly 30 % of the work time*”.

At Caelum, technical lunch, which is one of the practices adopted there to share knowledge across teams, is considered an effective practice for its referred purposes. It is shown through the effectiveness measure, which is composed by favourable values for its four components, as well as influenced by favourable conditions and stimuli.

7 The Enabling Context for Inter-team Knowledge Sharing

In this section, we present *theoretical integration* (Corbin and Strauss 2007), which is another step from grounded theory in which our proposed framework is discussed in the context of knowledge management literature. The part of this literature concerned with the personalization strategy has devoted attention to the fragile process of knowledge creation (Easterby-Smith and Lyles 2011; Dierkes et al. 2003). Knowledge creation is often subject to strong barriers, and depending on the context, some practices may work and be institutionalized and others may not (Krogh et al. 2000). Krogh and colleagues (2000) add that successful knowledge sharing process depends on the energy and sustained commitment an organization puts into knowledge creation.

In our case, we found out in the personalisation strategy employed by the investigated companies that the success of knowledge sharing across teams depends mostly on the organizational conditions and stimuli for leveraging interaction, along with KS practices (Nonaka and Takeuchi 1995; Nonaka 1988). These aspects are what (Krogh et al. 2000) call the enabling context, that is crucial for the knowledge-sharing effectiveness (Choo and Alvarenga 2010; Wang and Noe 2010).

Studies from the Learning Software Organizations (LSO) field bridge theories from the organizational learning field (e.g. Argyris (1999), Nonaka and Takeuchi (1995), and Wenger’s theories on Communities of Practice (Wenger et al. 2002)) to software organizations. Some of the studies are focused on employing practical knowledge management (Salo 2005; Holz and Melnik 2004; Doran 2004; Holz and Maurer 2003; 2002; Brössler 1999), pointing out the need to be careful on using tools and knowledge repositories to ensure that they do not become “experience cemeteries”. Holz and Melnik (2004) state that LSO consider learning as part of the organizational overall strategy, so it should be accompanied with explicit goals and methods, the appropriate use of structure, and systems to empower the interaction among people. Even for the software development field, which is characterized by systematizing processes with tools, our results relate to studies in organizational learning field, which highlight the importance of caring about the socio-technical factors that are crucial for the success of KS practices.

Pettersen et al. (2012) have also provided techniques for small and medium sized software companies to implement practical knowledge management. The authors suggest balancing the use of technology (knowledge repositories and project diaries), processes (project retrospectives, process guides, project memory cards, and client scene investigation), expertise networks (knowledge brokers, technical information meetings), and space (visual boards). These latter two techniques focus on connecting and facilitating interaction across teams, and fostering visibility within the team and to others who have an interest in the team’s work. The agile companies we investigated in our study have adapted several known intra-team practices, such as retrospectives, to be adopted also across teams.

Informative workspaces and face-to-face conversations are two practices that we identified as important in our study. This result is aligned with previous work from Pettersen et al. (2012) who suggests that few levels in hierarchy and few walls around the organizational structure (as advocated by Krogh et al. (2000)), along with a collaborative culture (as stated by Donate and Guadamillas (2011)), and a consistent strategy (also observed by Donate and Canales (2012)) impact on the easy access to face-to-face conversations and informative workspaces of different teams. Likewise, agile companies that take part in a very dynamic competitive environment, because of their business domain, embrace a proactive strategy (Donate and Canales 2012) to produce more innovation by adopting work practices, such as marathons with positive competitions, periodic technical presentations, and Coding Dojos. These are all practices adopted in the studied organizations.

After recognition of the theory of organizational knowledge creation, Nonaka and his associates expanded their work into a more general theory, in which the concept of *ba* and enabling conditions play a dominant role in organizational knowledge creation (Nonaka and von Krogh 2009; Nonaka and Toyama 2007; Dierkes et al. 2003; Nonaka et al. 2000; Nonaka and Konno 1998). The term *ba* consists of a physical, virtual, mental or any combination of these kinds of spaces in which knowledge is shared, created, shared, and utilized, since knowledge needs a context in order to exist. The main objective of *ba* is interaction. It needs to be ‘energized’ (stimulated) to become active and create meaning into the workspace.

Choo and Alvarenga (2010) made a comprehensive literature review around the concept of *ba*. They synthesized four groups of conditions that enable knowledge creation. They also analyzed enabling conditions by the types of knowledge processes and levels of interaction to address a particular knowledge problem or vision. These authors state that different conditions support different *ba* in different ways, and the task of KM is to arrange these factors in such a manner that the KM activities can be achieved as smoothly as possible.

We found out that our results in the context of agile software development organizations fit in Choo's and Alvarenga's theoretical framework. Due to lack of space, we discuss the influencing factors most referred in the literature in the subsections below.

7.1 Social/Behavioral Dimension

Choo and Alvarenga (2010) grouped in this category social relationships and interactions based on norms and values, such as trust, care, empathy, attentive enquiry and tolerance. Organizational culture is a crucial factor because it guides the organizational functioning and its members' thinking and behavior. Yet, as Pettigrew (1979) states, it is fundamental for putting the social structure of an organization together and for managing organizational change and renewal. In our study we observed that organizational culture and the individual behavior towards knowledge sharing are mutually related. Joia and Lemos (2010) explain this relation by reporting that both require a favorable environment for questioning, relationships based on trust between the individuals, development of a common language, and internal communication flow. We observed all these aspects in the organizations we studied.

Agile methods comprise a set of underlying values and principles that are in line with this category. Beck and Andres (2004) highlight that agile teams require courage, respect, communication, simplicity, and feedback. Likewise, Highsmith and Cockburn (2001) state values of common focus, mutual trust, and respect. However, on the organizational level, care must be taken for analyzing the organizational culture under the agile teams' culture.

The adoption of PP by different teams' members described in Section 5.5 is an example that outlines the relevance of the organizational culture. At Caelum and ThoughtWorks (Brazil) we experienced a collaborative organizational culture, as teams' members are more willing to share knowledge and to explain topics of interest to others. On the other hand, at UOL an interviewee stated that teams' members still feel uncomfortable when pairing with some people.

Even with several studies reporting positive effects of Pair Programming, such as the construction of shared knowledge in collaborative problem-solving, there are still some inconclusive results and also some studies even report negative effects (Hannay et al. 2009). Williams and Kessler (2002) stress the role of collaboration in PP by stating that "pair programming works when the pairs are tightly integrated, interacting and working closely. Recently, Plonka and van der Linden (2012) have identified aspects that hinder the amount of pairing in four companies. They outline that these aspects are related to organizational issues, such as organizational culture and management style, which are in line to what we found out in our study. As a consequence, organizational culture is considered fundamental to scaling PP across teams.

In a comprehensive study employed by Iivari and Iivari (2010) on organizational culture, these authors described that agile methods present some contradictions that may impact the absorption of the agile philosophy at the organizational level. They state that the agile methods ideology is similar to what they term as developmental culture⁵. However, they claim that there is a need for a reasonable balance between the types of culture in agile context, since agile methods also consider several aspects of the rational culture⁶, such as

⁵Developmental culture orientation: growth, resource acquisition, creativity and adaptation to the external environment (change and external focus) (Iivari and Iivari 2010).

⁶Rational culture orientation: productivity, efficiency and goal achievement (stability and external focus) (Iivari and Iivari 2010).

the adaptive and flexible response to the environmental volatility and the focus on values of the agile team culture, the timeboxed deadlines, productivity and goal achievement.

Although we have witnessed the four companies participating in this study following the agile ideology within teams, some of these companies, such as UOL (which used to adopt a prescriptive software development processes and traditional approaches to management), experienced that the organizational culture is acting as a barrier to the agile culture. Coplien and Harrison (2004) clarify that this aspect should be a concern when scaling agile methods, as the organizational culture can make agile adoption easier or harder.

Inter-team knowledge sharing is influenced by social barriers, such as reluctance of team members to waste their time or the time of others; reluctance to interrupt others; and unwillingness to expose or discuss problems. Whitworth (2006) reported these social barriers in agile teams. These barriers are often mitigated by promoting team awareness and feedback, by increasing investment and involvement in the collective endeavor, and by supporting frequent and informal face-to-face interaction, which are in line with what the studied companies have applied to minimize the social barriers.

7.2 Cognitive/Epistemic Dimension

In this dimension, Choo and Alvarenga (2010) classified the need for both knowledge diversity and common knowledge (which is based on shared beliefs and mental models). They claim that these requirements need to reinforce each other, since the existence of shared beliefs should be based on embracing the ideas and experiences of people with different backgrounds and perspectives.

The easy access to diverse conversations through the practices described in Section 5, such as face-to-face conversations, technical presentations, collective meetings, team member rotation, marathons, etc; and to boards or posters (informative workspace) in shared workspaces enables colleagues from different teams to get to know or to suggest existent solutions to others. As a consequence, the studied companies related the practices to the creation of a shared language in different teams. Shared language is an important aspect described by the agilist Cockburn (2006) to achieve collective knowledge. Way before this author, in the sociology of knowledge field, Berger and Luckmann (1967) stated that knowledge is socially constructed through shared experience. Likewise, the knowledge management advocates, Nonaka and colleagues (Nonaka and von Krogh 2009; Nonaka and Toyama 2007), refer to the importance of establishing the “organic concentration” of knowledge in the organizations. Thus, the practices described in this study (Section 5) are in line to what these authors advocate and showed to play a fundamental role in creating organizational common knowledge.

Salo (2005) investigated agile retrospectives as a way to improve and adapt software development processes. The author concluded that the organizational level could only benefit from the learning of project teams if the knowledge and reasoning behind the process improvements were converted into an explicit format (such as a structured template) that could also be used for learning at the organizational level. This is aligned with research on post-mortem reviews (Dingsøyr and Hanssen 2003). While making knowledge explicit from retrospectives is usually regarded as a good strategy for codification, our results suggest that most of the explicit knowledge recorded in tools and repositories were difficult to maintain and to be revisited by the studied agile teams. Because of that, these agile teams went a bit further, revealing that they absorb tacit knowledge dynamically through the participation of professionals from different teams in retrospectives and in other collective meetings as a way to cross-fertilize tacit knowledge.

The explanation of how team integration improves knowledge sharing is presented Easterby-Smith and Lyles (2011) through the process of cross-fertilization between teams (Nonaka and Takeuchi 1995), i.e., when team members are part of more than one team simultaneously.

Epistemic diversity, as an appreciated agile principle - “Business people and developers must work together daily throughout the project / The sponsors, developers, and users should be able to maintain a constant pace indefinitely” (Agile Manifesto for Agile software Development (Beck et al. 2001)) - is, in our study, mainly achieved through assigning people with different roles to adopt the discussed work practices.

The studied organizations benefit from combining the diverse knowledge of their members, since this combination impacts the companies’ outcomes through enhanced technical excellence, quality of products, performance, and innovation. Whitworth (2006) declared that sharing knowledge and receiving feedback allow a sense of common knowledge and collective action. As a consequence, a company becomes more prepared to respond to changes and adapt to a dynamic environment (Eckstein 2004). Enterprise agility is also achieved through the following main characteristics: (1) flexibility and adaptability, (2) responsiveness, (3) speed, (4) integration and low complexity, (5) mobilization of core competencies, (6), high quality and customized products, and (7) culture of change (Sherehiy et al. 2007). These authors state that organizational-level agility means to manage long- and short-term changes within the production system through cooperation. Cooperation across teams and/or departments places knowledge and ability of employees to deal with the turbulent market changes.

7.3 Strategy/Structure Dimension

This dimension is characterized by the organization’s need to provide direction and structure. Takeuchi and Nonaka (2004) argue that enabling knowledge creation has a close connection to organizational structure and strategies that affects all other influencing factors, for instance top management commitment to knowledge management initiatives.

Practices adopted by the studied companies, such as team member rotation described in Section 5.3, Coding Dojos (Section 5.8), and collective meetings (Section 5.4), adopted by the companies under study showed special reliance on organizational strategy, top management and leadership support leading to better integration among teams and projects.

Regarding developer rotation among teams and projects (Fægri et al. 2010), this needs to be adopted in a sustainable way and relies on the construction of an open and trustful organizational culture, and also on the overall alignment (strategy) to generate innovative learning. Again, we outline the importance of the organizational conditions present in the companies over the adoption of the knowledge sharing practices.

At Apontador, we could observe teams’ members striving to adopt Coding Dojos outside working hours (Section 5.8), and facing difficulties in keeping a constant pace and in increasing more participation of members. In this company, organizational strategy and top management and leadership support were not fostering the successful adoption of Coding Dojos. On the other had, at Caelum top management and leadership fostered the adoption of Coding Dojos considering its adoption as part of their organizational strategy. Because of that, management provided lunch for most of these sessions.

Face-to-face conversations (described in Section 5.1) and informative workspaces (described in Section 5.2) relied specially on the company’s physical structure. This aspect is raised by Eckstein (2004), as she points out that creating a learning environment in an

agile context requires fostering interaction among professionals through open plan offices and meetings, however, she highlight the intrinsic relationship with the organizational culture. Eckstein also exemplifies a meeting that would involve reviewing team members to define good and bad habits they detect most often in the company and present them through lectures, study groups, newsletters, and actions taken within projects (i.e., to designate focal points to eliminate bad habits for the next development cycle). In this scenario, she states that it is important to take into account the organizational culture when publishing the detected failures and bad habits. In companies that deal with failure as the best way to learn, this is no issue at all, these can be openly spoken. However, in companies with no established culture of failure, it is necessary to ensure that nobody will be able to detect which team is guilty of specific bad habits.

Caelum and ThoughtWorks (Brazil) are companies that deal with this culture of failure. Both companies provide cross-team level transparency to allow greater awareness of the problems, issues, common interests and/or objectives of others across teams, and a better understanding of their own opinions and roles in relation to the rest of the teams through open discussions.

Agile methods support cohesive teamwork through team commitment to a collective and clear goal: to de-liver the most business value to the customer in the shortest amount of time and with quality software code (Whitworth 2006). Likewise, the identified practices for socialization foster this commitment on an organizational or departmental level. In fact, the presence of a collective long-term strategy across agile teams was considered as a great driver for inter-team knowledge sharing. Detailed knowledge of what other teams were working on, gained through practices such as collective meetings, pair programming across different teams, and face-to-face conversations in the workspace, was seen to support a high level of informal and opportunistic collaboration. This effort was particularly true in teams committed to a collective goal (Whitworth and Biddle 2007; Eckstein 2004).

Regarding shared leadership through the whole team principle (Beck, 2004), our results describe major leadership support for inter-team knowledge sharing initiatives. However, depending on the organizational hierarchy type, top management support may not be so keen on these initiatives. An environment of equal partnership involving non-hierarchical shared leadership and responsibilities, and thus self-regulating teams is suggested by Whitworth (2006).

Eckstein (2004) and Cockburn (2006) have devoted great attention to shared workspaces, such as open plan offices or “war rooms” (type of physical structure that places team members in the same room in the company), as a way to provide more osmotic communication⁷. With an open workspace, we observed osmotic communication being scaled to other teams. Whitworth (2006) also states the importance of a large open plan room, with workstations set up, so that developers can work in a shared environment. Small cubicles are made available in case team members need privacy to work, but the bulk of activity occurs in the common space. Krogh et al. (2000) report that open workspaces reflect the new logic of knowledge intensive organizations, facilitating the continuous exposition to all kinds of operations of the organization.

⁷Osmotic communication means information flow into the background hearing of members of the team that makes people pick up relevant information as though by osmosis (Cockburn 2006).

7.4 Information Systems Dimension

In this dimension, the use of information systems and information management processes is analyzed to support knowledge management activities. Agile methods prescribe recursive human action and interaction in a way, which is heavily mediated by tools and processes. These characteristics led to the definition of agile teams as complex adaptive socio-technical systems (Whitworth and Biddle 2007; Baxter and Sommerville 2011).

Agile methods put a strong emphasis on constant communication and coordination between team members, particularly face-to-face interaction (Beck et al. 2001). Likewise, we have observed communication and coordination across teams occurring similarly through face-to-face conversations, collective meetings, collective informative workspaces, tools or frameworks to support automated tests, integration tests, continuous deployment, etc (Coplien and Harrison 2004).

The studied agile companies often experience difficulties in managing their explicit knowledge in tools or repositories (e.g., internal company tools, intranet, wikis, mailing lists, and blogs). As described in the beginning of Section 5, at UOL most teams fail to document their knowledge. These difficulties are also explained in the codification strategy studies presented in Section 2.

Socio-technical systems follow the growing trends in research and theory from information systems (IS), computer supported cooperative work (CSCW), and operations management, and are in line with the common conceptualization in psychology of work teams as complex adaptive systems (Ilgen et al. 2005). This concept takes into consideration several factors, such as systems comprised of people, machines, and methods organized to collect, transmit, and disseminate information (Whitworth and Biddle 2007). Knowledge sharing initiatives are in line with socio-technical systems, as they need to foster an organic dissemination of knowledge, so that it becomes a sustainable endeavor. For instance, Caelum team members consider they employ this organic dissemination of knowledge since this company applies the set of practices described in Section 5, consider they employ this organic dissemination of knowledge. Finally, it is important to stress that the adoption of KS practices needs to be considered along with organizational conditions and stimuli to compose an enabling context.

7.5 Integrating our Main Contribution with the Previously Mentioned Dimensions

Recalling our research question: *How organizations achieve knowledge sharing effectiveness across agile teams using personalization strategies?* We highlight that the practices by themselves may not be sustained in a company, that is why companies must be aware that they need to establish an enabling context, considering other important factors for keeping the KS endeavor sustainable.

Back to the conceptual model, in Fig. 5 we outline in blue the relationship between our conceptual model and the theoretical framework depicted by Choo and Alvarenga (2010). Our theory relates to the *enabling conditions* in four groups of factors for the *knowledge sharing process* in the *organizational level*.

In the cube of Fig. 5, we present one example of how we categorized enabling conditions of our conceptual model, such as Strategy, Structure, Top management and leadership support, Size, Integration among teams and projects, KS Practices, and Stimuli (sustainable pace and incentives) within the Strategy/Structure dimension. But, in fact, in the previous sections we discuss our results in the context of all dimensions.

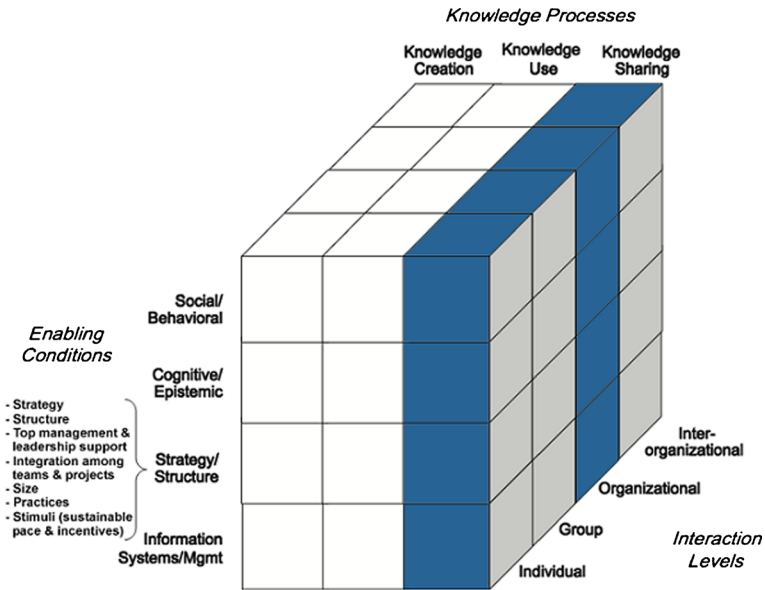


Fig. 5 Relationship between our conceptual model and the theoretical framework proposed by Choo and Alvarenga (2010)

8 Limitations

In this section we describe the limitations of our research. First of all, we had some limited experience in conducting this empirical study, because in three organizations it was not possible to observe the daily work of practitioners in an extensive way. This happened because of several reasons. Firstly, because of the distance (one of the organizations was located in another Brazilian state, around 1,150 km far from the state of São Paulo). Secondly, because of confidentiality matters, and lastly, because the organizations were afraid that the researchers could influence the practitioners’ work. In these last two cases, interviews and semi-structured questionnaires constituted the major source of data.

Observations, in most companies, were undertaken under confidentiality agreements (Non-Disclosure Agreement - NDA) and after an invitation. In other words, observation took place only when the informants invited the authors to participate in meetings or other events. Only Caelum, due to its business domain, gave us the freedom to observe the everyday work without any confidentiality agreement and predefined agenda, so we could arrive there at anytime.

Considering the researchers’ expertise on the field, we did not start the research without any prior knowledge from the theme and the organizations in study. However, our knowledge and experience on the research method helped us to avoid that our knowledge of the field would lead us to preconceived theoretical ideas. These ideas could hinder the emergence of ideas firmly linked to the data.

9 Final Remarks

From the analysis of four Brazilian agile software organizations and one specialist in agile methods implementation, this study seeks to advance our knowledge of how agile software organizations can achieve inter-team knowledge sharing effectiveness.

Our main contribution is a conceptual model that seeks to understand this phenomenon by explaining that effectiveness is related to applying purposeful practices, along with organizational conditions and stimuli. This model fosters awareness that inter-team knowledge sharing effectiveness cannot only be reached by a list of specific practices, but by a sustainable context in which the whole process is implemented.

Agile software development is now moving onto the next level, scaling agility to the organizational level. Cross-team knowledge sharing informs this endeavor and also reflects the way agile software organizations are coping with long-term strategies and the consideration of knowledge as a resource for organizational competitiveness.

In further research, we intend to refine, extend and validate the conceptual model by exploring the effectiveness of inter-team knowledge sharing practices in other knowledge enabling contexts. Other validations should be accomplished in future studies regarding the intricate relationships between the proposed conditions and stimuli, the proposed knowledge sharing purposes and the practices applied specifically in agile software development.

Acknowledgments This work was supported by FAPESP project no. 2009/16354-0 and CNPQ project no. 476661/2010-2. We are grateful to all organizations and members participating in the study.

References

- Altheide DL, Johnson JM (1994) Criteria for assessing interpretive validity in qualitative research. In: Handbook of Qualitative Research, Sage Publications, pp 485–499
- Anice M, Silveira G (2011) Increasing learning in an agile environment: Lessons learned in an agile team. In: Proceedings of the Agile Conference, Salt Lake City, UT, pp 289–295
- Argyris C (1999) On organizational learning. Blackwell Publishing
- Baxter G, Sommerville I (2011) Socio-technical systems: from design methods to systems engineering. *Interact Comput* 23:4–17
- Beck K (2002) Test-driven development by example. Addison Wesley
- Beck K, Andres C (2004) Extreme programming explained: embrace change, 2nd edn. Addison-Wesley
- Beck K, Beedle M, van Bennekum A, Cockburn A, Cunningham W, Fowler M, Grenning J, Highsmith J, Hunt A, Jeffries R, Kern J, Marick B, Martin RC, Mellor S, Schwaber K, Sutherland J, Thomas D (2001) Principles behind the agile manifesto. <http://agilemanifesto.org/principles.html>
- Berger PL, Luckmann T (1967) The social construction of reality: a treatise in the sociology of knowledge. Anchor Books, Garden City
- Bjørnson FO (2008) Knowledge management in software engineering: a systematic review of studied concepts, findings and research methods used. *Inform Softw Technol* 50:1055–1068
- Boden A, Avram G, Bannon L, Wulf V (2009) Knowledge management in distributed software development teams does culture matter? In: Global Software Engineering, Fourth IEEE International Conference on, IEEE, pp 18–27
- Brössler P (1999) Knowledge management at a software engineering company an experience report. In: Proceedings of the International Workshop on Learning Software Organizations (LSO), Springer, pp 77–86
- Brown JS, Duguid P (1991) Organizational learning and communities-of-practice: Toward a unified view of working, learning and innovation. *Organization Sci* 2(1)
- Cabral AY, Ribeiro MB, Lemke AP, Silva MT, Cristal M, Franco C (2009) A case study of knowledge management usage in agile software projects. In: International Conference on Enterprise Information Systems (ICEIS), pp 627–638

- Chau T (2005) Inter-team learning for agile software processes. Master thesis, University of Calgary, Department of Computer Science, Canada
- Chau T, Maurer F (2004) Knowledge sharing in agile software teams. In: *Lecture Notes in Computer Science*, Springer-Verlag Berlin Heidelberg, pp 173–183
- Chau T, Maurer F (2010) Knowledge sharing in agile software teams. *Logic Approx* 3075:173–183
- Chau T, Maurer F, Melnik G (2003) Knowledge sharing: agile methods vs. Tayloristic methods. In: Press ICS (ed) 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), pp 302–307
- Choo CW, Alvarenga RCD (2010) Beyond the ba: managing enabling contexts in knowledge organizations. *J Knowl Manage* 14(4):592–610
- Cockburn A (2006) *Agile software development: the cooperative game* (2nd Edition). Addison-Wesley Professional
- Conboy K (2009) Agility from first principles: reconstructing the concept of agility in information systems development. *Inform Syst Res* 20(3):329–354
- Conboy K, Fitzgerald B (2010) Method and developer characteristics for effective agile method tailoring: a study of xp expert opinion. *ACM Trans Softw Eng Methodol* 20(1)
- Coplien JO, Harrison NB (2004) *Organizational patterns of agile software development*. Prentice Hall
- Corbin J, Strauss AC (2007) *Basics of qualitative research: techniques and procedures for developing grounded theory*, 3rd edn. Sage Publications
- Corbucci H, Goldman A, Katayama E, Kon F, Melo CO, Santos V (2011) Genesis and evolution of the agile movement in Brazil - a perspective from the academia and the industry. In: *Proc. of 25th Brazilian Symposium on Software Engineering (Track: SBES is 25)*, pp 98–107
- Cummings JN (2004) Work groups, structural diversity, and knowledge sharing in a global organization. *Manage Sci* 50(3):352–364
- Derby E, Larsen D (2006) *Agile retrospectives - making good teams great*. Pragmatic Bookshelf
- Desouza KC (2003) Facilitating tacit knowledge exchange. *Commun ACM* 46(6):85–88
- Dierkes M, Antal AB, Child J, Nonaka I (2003) *Handbook of organizational learning and knowledge*. Oxford Press
- Dingsøyr T, Hanssen G (2003) Extending agile methods: postmortem reviews as extended feedback. In: Henninger S, Maurer F (eds) *Advances in Learning Software Organizations*. 4th international Workshop LSO 2002. Springer Verlag, Berlin Heidelberg, pp 4–12
- Dingsøyr T, Nerur S, Balijepally V, Moe N (2012) A decade of agile methodologies: towards explaining agile software development. *J Syst Softw* 85(6):1213–1221
- Donate MJ, Canales JI (2012) A new approach to the concept of knowledge strategy. *J Knowl Manage* 16(1):22–44
- Donate MJ, Guadamillas F (2011) Organizational factors to support knowledge management and innovation. *J Knowl Manage* 15(6):803–814
- Doran HD (2004) Agile knowledge management in practice. In: LNCS 3096. *Advances in Learning Software Organizations*. In the 6th International Workshop, LSO., pp 137–143
- Du M (2007) Knowledge management: what makes complex implementations successful? *J Knowl Manage* 11(2):91–101
- Easterby-Smith M, Lyles MA (2011) *Handbook of organizational learning and knowledge management*, 2nd edn. Wiley
- Eckstein J (2004) *Agile software development in the large: diving into the deep*. Dorset House
- Fægri TE (2009) Improving general knowledge in agile software organizations - experiences with job rotation in customer support. In: *The Agile Conference*, pp 49–56
- Fægri TF, Dybå T, Dingsøyr T (2010) Introducing knowledge redundancy practice in software development: experiences with job rotation in support work. *Inform Softw Technol* 52:1118–1132
- Fraser S, Rising L, Ambler S, Cockburn A, Eckstein J, Hussman D, Miller R, Striebeck M, Thomas D (2006) A fishbowl with piranhas coalescence, convergence or divergence: The future of agile software development practices some assembly required! In: *Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications OOPSLA*, pp 937–939
- Glaser BG, Strauss AL (1967) *The discovery of grounded theory: strategies for qualitative research*. Aldine Publishing Company, New York
- Guba E, Lincoln YS (1994) Competing paradigms in qualitative research. In: *Handbook of Qualitative Research*, Sage Publications, pp 105–117
- Guba EG, Lincoln YS (1989) *Fourth generation evaluation*. Newbury Park, CA, Sage
- Hannay J, Dyba T, Arisholm E, Sjøberg D (2009) The effectiveness of pair programming: a meta-analysis. *Inform Softw Technol* 51(7):1110–1122

- Hansen MT, Nohria N, Tierny T (1999) What's your strategy for managing knowledge? *Harvard Business Rev* 77:106–116
- Highsmith J, Cockburn A (2001) Agile software development: the people factor. *IEEE Comput* 34(11):131–133
- Holz H, Maurer F (2002) Knowledge management support for distributed agile software processes. In: *Proceedings of the Workshop on Learning Software Organizations (LSO)*, pp 60–80
- Holz H, Maurer F (2003) Knowledge management support for distributed agile software processes. *LNCS, Learn Softw Organ (LSO)* 2640:60–80
- Holz H, Melnik G (2004) Research on learning software organizations - past, present and future. In: *Advances in Learning Software Organizations. In the 6th International Workshop, LSO*, pp 1–6
- Holz H, Schafer J (2003) Collaborative, task-specific information delivery for agile processes. In: *Society IC (ed) 12th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp 320–325
- Holz H, Melnik G, Schaaf M (2003) Knowledge management for distributed agile processes: Models, techniques, and infrastructure. In: *Press ICS (ed) 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, pp 291–294
- Iivari J, Iivari N (2010) Organizational culture and the deployment of agile methods: the competing values model view. *Springer*, pp 203–222
- Ilgel DR, Hollenbeck JR, Johnson M, Jundt D (2005) Teams in organizations: from input-process-output models to imoi models. *Ann Rev Psychol* 56:517–543
- Joia LA, Lemos B (2010) Relevant factors for tacit knowledge transfer within organisations. *J Knowl Manage* 14(3):410–427
- Kähkönen T (2004) Agile methods for large organizations - building communities of practice. In: *Agile Development Conference, IEEE*, pp 2–11
- Karlsen JT, Hagman L, Pedersen T (2011) Intra-project transfer of knowledge in information systems development firms. *J Syst Inform Technol* 13(1):66–80
- Kavitha RK (2011) A knowledge management framework for agile software development teams. In: *International Conference on Process Automation, Control and Computing (PACC)*, pp 1–5
- Keplinger W (2007) Agility in information systems development: characterisation, motivation and conceptualisation. *iSChannel - Inform Syst Stud J* 2:25–28
- Kettunen O (2010) Agile product development and strategic agility in technology firms. Master thesis, Helsinki University of Technology, Finland
- Kettunen P, Laanti M (2008) Combining agile software projects and large-scale organizational agility. *Softw Process Improv Pract* 13(2):183–193
- Krogh GV, Ichijo K, Nonaka I (2000) *Enabling knowledge creation: how to unlock the mystery of tacit knowledge and release the power of innovation*. Oxford University Press, Oxford
- Levy M, Hazzan O (2009) Knowledge management in practice: the case of agile software development. In: *ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pp 60–65
- Lincoln YS, Guba EG (1985) *Naturalistic inquiry*. Sage, Newbury Park
- Lindvall M, Basili VR, Boehm BW, Costa P, Dangle K, Shull F, Tesoriero R, Williams LA, Zelkowitz MV (2002) Empirical findings in agile methods. In: *Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe*
- Lyytinen K, Rose GM (2007) Information system development agility as organizational learning. *Eur J Inform Syst* 15:183–199
- Maalej W, Happel HJ (2008) A lightweight approach for knowledge sharing in distributed software teams. *7th International Conference on Practical Aspects of Knowledge Management (PAKM)* pp 14–25
- Maranzato RP, Neubert M, Herculano P (2011) Moving back to scrum and scaling to scrum of scrums in less than one year. In: *The ACM international conference companion on Object oriented programming systems languages and applications companion (SPLASH)*, pp 125–130
- Merriam SB (2002) Assessing and evaluating qualitative research. In: *Assessing and evaluating qualitative research in practice*, San Francisco, CA, Jossey-Bass, pp 18–33
- Mestad A, Dingsøy T, Dybå T (2007) Building a learning organization: Three phases of communities of practice in a software consulting company. In: *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS)*
- Nerur S, Mahapatra R, Mangalaraj G (2005) Challenges of migrating to agile methodologies. *Commun ACM* 48(5):72–78
- Neves FT, Correia AMR, Rosa VN, Neto MC (2011) Knowledge creation and sharing in software development teams using agile methodologies: key insights affecting their adoption. In: *6th Conferência Ibérica de Sistemas e Tecnologias de Informação*

- Nonaka I (1988) Creating organizational order out of chaos: self-renewal in Japanese firms. *Calif Manag Rev* 30:57–73
- Nonaka I, Konno N (1998) The concept of ba: building a foundation for knowledge creation. *Calif Manag Rev* 40:40–54
- Nonaka I, von Krogh G (2009) Tacit knowledge and knowledge conversion: controversy and advancement in organizational knowledge creation theory. *Organ Sci* 20(3):635–652
- Nonaka I, Takeuchi H (1995) *The knowledge-creating company: how Japanese companies create the dynamics of innovation*. Oxford University Press, Oxford
- Nonaka I, Toyama R (2007) Strategic management as distributed practical wisdom (phronesis). *Indust Corp Change* 16:371–394
- Nonaka I, Toyama R, Konno N (2000) Seci, ba and leadership: a unified model of dynamic knowledge creation. *Long Range Plan* 33(1):5–34
- Pettersen R, Michael M, Monfils FF, Dingsøyr T, Saadaoui S, Bjørnson FO, Neophytou K, Hadjioannou A (2012) *Practical knowledge management - Techniques for small and medium sized companies*, 1st edn. EXTRA Consortium
- Pettigrew AM (1979) On studying organizational cultures. *Administrative Sci Quart* 40:570–581
- Pikkarainen M, Salo O, Still J (2005) Deploying agile practices in organizations: A case study. In: Heidelberg SVB (ed) *EuroSPI*, pp 16–27
- Pikkarainen M, Haikara J, Salo O, Abrahamsson P, Still J (2008) The impact of agile practices on communication in software development. *Empirical Softw Eng* 13(3):303–337
- Plonka L, van der Linden J (2012) Why developers don't pair more often. In: CHASE, pp 123–125
- Qureshi MRJ, Kashif M (2009) Seamless long term learning in agile teams for sustainable leadership. In: 5th IEEE International Conference on Emerging Technologies, pp 389–394
- Rech J, Bogner C (2010) Qualitative analysis of semantically enabled knowledge management systems in agile software engineering. *Int J Knowl Manage* 6(2):66–85
- Runeson P, Höst M (2009) Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw Eng* 14(2):131–164
- Salo O (2005) Systematical validation of learning in agile software development environment. *LNAI, Learn Softw Organ (LSO)* 3782:106–110
- Salo O, Abrahamsson P (2007) An iterative improvement approach for agile development: implications from multiple case study. *Softw Process Improve Pract* 12(1):81–100
- Santos V, Goldman A, Shinoda AC, Fischer A (2011) A view towards organizational learning: An empirical study on scrum implementation. In: 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE), pp 583–589
- Santos V, Goldman A, Santos C (2012) Uncovering steady advances for an extreme programming course. *CLEI Electron J* 15(1):1–20
- Sato DT, Corbucci H, Bravo MV (2008) Coding dojo: an environment for learning and sharing agile practices. In: *Proceedings of the Agile Conference*, pp 459–464
- Schwaber K (2004) *Agile project management with scrum*. Microsoft Press
- Schwaber K, Beedle M (2002) *Agile software development with SCRUM*. Prentice-Hall
- Schwandt TA (2007) *Dictionary of qualitative inquiry*, 3rd edn. Sage, Thousand Oaks
- Shalloway A, Beaver G (2009) *Trott JR, Lean-agile software development: achieving enterprise agility*. Addison-Wesley Professional
- Sherehiy B, Karwowski W, Layer J (2007) A review on enterprise agility: concepts, frameworks, and attributes. *Int J Industr Ergonom* 37:445–460
- Silva MAC, Filho HR, Silva HFN (2010) Analysis of ba during scrum process (in portuguese). In: *Proc. of 17th Symposium of Production Engineering (SIMPEP)*. Proj. Management and Production Engineering. Bauru, Sao Paulo, Brazil, pp 1–15
- Sommerville I (2010) *Software engineering*, 9th edn. Addison-Wesley
- Srinivasan J, Lundqvist K (2009) Using agile methods in software product development: a case study. In: *Sixth International Conference on Information Technology: New Generations*
- Stettina CJ, Heijstek W, Fægri TE (2012) Documentation work in agile teams: the role of documentation formalism in achieving a sustainable practice. In: *Agile Conference (AGILE)*, pp 31–40
- Sveiby KE (2004) *People-focused knowledge management: how effective decision making leads to corporate success*. Butterworth-Heinemann
- Takeuchi H, Nonaka I (2004) *Hitotsubashi on knowledge management*. Wiley
- Wang S, Noe RA (2010) Knowledge sharing: a review and directions for future research. *Human Resour Manage Rev* 20:115–131
- Wang X, Maurer F, Morgan R (2010) Tools for supporting distributed agile project planning. *Agility Across Time and Space*, pp 183–200

- Wenger E, McDermott R, Snyder W (2002) *Cultivating communities of practice: a guide to managing knowledge*. Harvard Business School Press, Boston
- Whitworth E (2006) *Agile experience: communication and collaboration in agile software development teams*. Master dissertation of the Department of Psychology, Carleton University Ottawa, Ontario, Canada
- Whitworth E, Biddle R (2007) The social nature of agile teams. Agile Conference (AGILE), Washington, DC, pp 26–36
- Williams L, Kessler R (2002) *Pair programming illuminated*. Addison-Wesley Longman Publishing Co., Inc, Boston



Viviane Santos received her M. Sc. in Computer Science from the State University of Ceará, Brazil, and her doctorate in Computer Science from the University of São Paulo, Brazil. She is currently a postdoctoral researcher at the University of São Paulo and an associate professor at the Federal University of Pará in Brazil. Her research interests include software engineering, software development methodologies, agile methods, organizational learning, and knowledge management.



Alfredo Goldman received his B.Sc. in Applied Mathematics from University of São Paulo, Brazil, his M.Sc. in Computer Science also from University of São Paulo, and his doctorate in Computer Science from the Institut National Polytechnique de Grenoble, France. He currently is an associated professor in the Department of Computer Science at University of São Paulo. His research interests include agile software development, and parallel and distributed computing. He has published over 100 papers on these areas.



Cleidson R. B. de Souza received his B.Sc. in Computer Science from the Federal University of Pará, his M.Sc. in Computer Science from the State University of Campinas and his PhD in Information and Computer Sciences from the University of California, Irvine. He is a researcher at the Vale Institute of Technology and an associate professor at the Federal University of Pará in Brazil. His research interests are in the field of computer-supported cooperative work as applied to software engineering, i.e., he is interested in understanding how software engineers work together to develop software. He adopts different methods in his research including ethnographies, surveys, and tool development and evaluation.