Technical Report: RT-MAC-2012-02
April, 2012

Institute of Mathematics and Statistics
University of São Paulo

FOSTERING INTER-TEAM KNOWLEDGE SHARING EFFECTIVENESS IN AGILE SOFTWARE DEVELOPMENT

VIVIANE SANTOS[a], ALFREDO GOLDMAN[b]

Department of Computer Science, Institute of Mathematics and Statistics, University of São Paulo,
Rua do Matão, 1010,
São Paulo, São Paulo, 05508-090, Brazil
{vsantos, gold}@ime.usp.br
http://ime.usp.br/~vsantos

AND

CLEIDSON R. B. DE SOUZA[b]

Vale Institute of Technology for Sustainable Development, ITV-DS
Tv. Dr. Moraes, 78, Nazaré
Belém, Pará, 66035-080, Brazil,
cleidson@cdesouza.net
https://www.cdesouza.net

Received (Day April 2012)
Revised (Day Month Year)
Accepted (Day Month Year)

Agile methods have impacted on software development by providing several changes on how software is developed. Agile values, principles and practices foster intra-team knowledge sharing. However, they do not inform how to cope with that across organizational levels. This study seeks to understand inter-team knowledge sharing effectiveness in agile software development organizations that employ practices for this endeavor. Using grounded theory, we analyzed data from four Brazilian agile software organizations and got feedback from an expert in agile methods implementation. Our main contribution is a conceptual model that explains how effectiveness depends on applying purposeful practices, along with organizational conditions and stimuli, that foster knowledge sharing across agile software development teams. This understanding represents an innovative focus within agile context and provides what to consider when striving on this path. Yet, it yields opportunities for further studies in refining and extending the model to several organizational contexts. Scaling agility toward organizational level is the next horizon for agile software development. Cross-team knowledge sharing leverages this endeavor and also reflects the way agile software organizations are coping with enterprise agility and the consideration of knowledge as a resource for organizational competitiveness.

Keywords: Agile software development; Conceptual model; Inter-team knowledge sharing effectiveness; Organizational conditions; Stimuli; Knowledge sharing practices; Grounded theory.

1. Introduction

[a] This is the present address of the author.
[b] This is the permanent address of the author.

Agile software development (ASD) is a lightweight paradigm for software development in response to the issues of the prescriptive methodologies, such as the waterfall approach. ASD focus on embracing changes and employs delivery of working software in iterative and incremental cycles, considering the human aspects of this endeavor, such as communication, collaboration, and coordination within a dynamic team environment [Karlsen et al, 2011].

The knowledge-intensive aspect of software development [Ruhe and Bomarius, 2000] encompasses technical, technological, and contextual knowledge [Chau and Maurer, 2004]. Hence, knowledge management (KM) initiatives are necessary to systematically leverage and reuse knowledge of the organization [Nonaka and Takeuchi, 1995].

For knowledge-based organizations, 90 percent of the knowledge is embeddedly and synthesizedly tacit [Wah, 1999]. Thus, for it to become an effective source of sustainable competitive advantage, it is essential to share tacit knowledge within the company [Wagner and Sternberg, 1999]. Knowledge sharing is one of the most important processes of KM, which gradually evolves and improves the production system and its elements. As a result, knowledge sharing is closely related to long-term performance and the competitiveness of an organization [Du, 2007].

Knowledge sharing in ASD is a very important task, due to its short, iterative nature and low documentation [Boden et al., 2009] [Cockburn, 2006] [Chau and Maurer, 2004]. Agile methods, such as eXtreme Programming (XP) [Beck and Andres, 2004] and Scrum [Schwaber, 2004], foster interaction and emphasize face-to-face tacit knowledge sharing within the team and between customers.

Even though agile methods can be very effective in software development, these methods strongly focus on empowering the project team in achieving its goals [Dingsøyr et al., 2010]. Little attention is given to creating insights and experience to the organizational level [Kettunen, 2009].

The excessive focus on the product and delivery of value to customer, lack of practice, pressure of time or the absence of formal knowledge-transfer system can make knowledge sharing difficult and, further, may lead to repeat past mistakes, rather than learn from experiences [Karlsen et al., 2011]. Therefore, there is a challenge to overcome the barriers to scale the knowledge on the group level to the organizational level effectively [Karlsen et al., 2011][Santos and Goldman, 2011][Bjørnson and Dingsøyr, 2008][Chau and Maurer, 2004] [Kähkönen, 2004].

Given this context, the research question guiding this enquiry is: “How inter-team knowledge sharing effectiveness may be achieved in agile software organizations?”. To answer this question, we employed a qualitative study using grounded theory [Glaser and Strauss, 1967] for analyzing data collected from four Brazilian agile software organizations that employ practices for this endeavor and gathered feedback from an expert in agile methods implementation. We considered grounded theory appropriate to our investigation because after a review of literature, we identified few studies on inter-team knowledge sharing in ASD. Yet, the research method’s emphasis on the learning

process allows for the emergence of original and rich findings rooted in the data, rather than out of existing theoretical models.

This study innovates by providing a conceptual framework that considers organizational conditions, stimuli and practices for effective inter-team knowledge sharing in ASD. Our main contribution is to explain how the influencing factors impact this process in agile software organizations. This means a next step of sharing knowledge in ASD, since such approach is considered effective in intra-team knowledge sharing. By enhancing ASD with support for organizational learning, the use of agile methods can be extended to a larger context [Shalloway et al., 2009].

This paper is organized as follows: Section 2 presents a literature review of knowledge sharing approaches in agile software development; Section 3 describes the research method; Section 4 reports the results; Section 5 contains a discussion on the results; and Section 6 presents the final considerations and further work.

2. Knowledge sharing in agile software development

Knowledge sharing is the provision of task information and know-how to collaborate with others to solve problems, develop new ideas, or implement policies or procedures [Cummings, 2004]. It can occur through technical solutions, known as codification strategy, or face-to-face communications, known as personalization strategy [Hansen et al., 1999] [Sveiby, 2004]. Tacit knowledge is especially shared through this second approach and demands organizations to develop platforms for facilitating knowledge sharing [Sirkemaa, 2008].

A relevant aspect of knowledge sharing is that the knowledge seeker has to get in touch with the knowledge provider, and the knowledge provider should be willing to share knowledge with the knowledge seeker. Seeking information from others in the organization may be considered as admitting ignorance on a given issue. To overcome these obstacles and benefit from this endeavor, it is important to facilitate and empower person-to-person tacit knowledge [Wang and Noe, 2010].

Since overall software development knowledge remains tacit in the developers' minds, generally team members are not aware of experiences acquired or successful solutions adopted from other teams within the organization [Brössler, 1999]. Therefore, they are often prone to either repeat past mistakes or duplicate effort in "re-inventing the wheel"[Basili et al., 2001].

Knowledge sharing in ASD is a key aspect due to its fundamental assumptions of delivering high-quality valuable software (release or increments) in short iterations, on tacit knowledge built up within the project team through face-to-face conversations, on embracing changes to provide competitive advantage to customer, and on cross-functional teams [Cockburn, 2004]. Agile methods have embedded enablers that foster knowledge sharing and refer specifically to the organizational changes required in migrating to agile [Levy and Hazzan, 2009]. This means employing agile values, principles and practices, which initiate several changes needed for a KM endeavor and

are critical to sustain an ongoing culture of knowledge sharing [Santos et al., 2011a] [Nerur et al., 2005].

2.1. *Intra-team knowledge sharing*

An important requirement for ASD is to make cross-functional, collocated team. This relevant aspect was raised in the Agile Manifesto [Beck et al., 2001], and it integrates into the team all needed roles in a whole team concept with small teams and collocation to minimize communication problems. Other authors outline that functionally diversified team positively influences knowledge sharing [Wang and Noe, 2010] [Von Krogh et al, 2000] [Nonaka and Takeuchi, 1995].

Agile values and principles contribute to changes in team members attitudes and to strengthen relationships. It happens due to increase of trust, communication and transparency in the relationships among team members and customers, less imposition by managers, greater sense of commitment, reciprocity, responsibility, organization and freedom of expression (questionings, opinions, etc.). Also due to better understanding of the work processes, and decrease of individual dependence through collective code ownership and self-organizing teams [Karlsen et al., 2011][Santos and Goldman, 2011] [Levy and Hazzan, 2009][Qureshi and Kashif, 2009].

Agile methods also recommend intra-project ceremonies like daily stand-up meetings, planning meetings, review meetings, and retrospectives. These meetings encourage interaction to report on work progress, to deal with impediments and obstacles, to provide visibility and opportunity to perform, to establish work evaluation and improvements, and to bring up transparency within the team.

Specifically for XP, the practices of pair programming (PP) and pair rotation play an important part in creating and reinforcing team relationship networks, and in disseminating mostly tacit knowledge within the entire team. Also, the informative workspace provides internal and external project visibility besides facilitates communication. In addition, the on-site customer fosters learning and feedback between the customers and the team [Chau, 2005].

Intra-team knowledge sharing is important for accomplishing specific project tasks and it offers the opportunity to discover creative means to improve organization's competitiveness. The agile culture ends up nurturing a natural environment for establishing approaches for sharing knowledge within the team, for instance [Treccani and Souza, 2011] present an empirical study of refactoring activities in agile software organizations, which are conducted in a collaborative way.

2.2. *Inter-team knowledge sharing*

Inter-team knowledge sharing is a significant aspect and a step towards creating organizational knowledge. In ASD multiple inter-dependent teams, or teams using similar technologies and processes, or multiple teams that face common problems or have overlapping interests in specific knowledge areas are more susceptible to share

knowledge [Karlsen et al., 2011]. Hereafter we report some attempts presented in literature to address this endeavor.

Project members rotation is a common practice of transferring workers from one project team to another by encouraging knowledge leveling and redundancy [Fægri et al., 2010]. However, the authors state this is a challenging practice, since it incurs a collective cost that must be amortized and legitimized by the organization, and yet can expose and confront implicit organizational values.

Scrum of scrums are regularly short meetings to coordinate work of cross-project sub-teams involving senior people or common technical areas [Schwaber and Beedle, 2002]. This practice is more intended 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, and Scrum masters need to have a broad view of products.

Communities of practice (CoP) are a mechanism for sharing knowledge in self-organizing informal cross-team communities [Wenger et al., 2002] [Brown and Duguid, 1991]. Some authors reported the use of CoP in agile environments [Mestad et al., 2007] [Kähkönen, 2004] and stated that, considering the dynamic and human centric approach of ASD, the standard form of CoP have to be adapted to affect agile teams, so a flexible model for organizing knowledge sharing, like dynamic skill circles with voluntary participation and freedom to participate in several communities, seems to provide greater learning effect.

Likewise, Birds of a feather[c] (BoF) session is a variation of CoP consisting ofan informal discussion group formed in an ad-hoc manner. This practice is usually held on agile methods conferences [Conboy and Fitzgerald, 2010] and should be adapted and tailored to work well in agile software organizations for the purpose of sharing knowledge [Santos and Goldman, 2011b].

[Desouza, 2003] argue that effective knowledge sharing requires, in first place, getting people motivated to talk and share their know-how. As knowledge is generated by people at the individual level, the author proposes organizational rooms and space for sharing knowledge, such as game and coffee rooms, instead of technical solutions.

Open fishbowl discussions[d] are a disciplined form of dialog used to discuss shared interest topics within large groups. This practice was firstly held on Object Oriented Programming,Systems, Languages and Applications (OOPSLA) conferences [Fraser et al., 2006], and is usually held on agile methods conferences. This disciplined approach consists of five chairs arranged in front of the room or in a circle (surrounded by the audience), so people interested in reporting anything about the topic under discussion in the session could sit down and express their opinions. The discussions can occur only between people sitting in chairs. The rest of the room should be silent, without any parallel conversation. Among the five seats, there should be always a free chair for the next person interested in discussing the topic. Hence, if someone seats in the remaining

[c]http://en.wikipedia.org/wiki/Birds_of_a_Feather_(computing).
[d]http://en.wikipedia.org/wiki/Fishbowl_(conversation)

empty chair, one of the four seats would release its own chair, as he/she already laid out his/her vision. This practice allows for disciplined interaction among participants to solve problems, discuss how to improve or reuse solutions, and raise feedback [Santos et al., 2011b].

Even with the report of the above practices intended to share knowledge among teams, few organizations actually apply them in broader purpose. Still, they are not effective in all contexts and must be carefully adapted and tailored [Santos and Goldman, 2011a].

3. Settings

This study aims at understanding inter-team knowledge sharing effectiveness in agile software development organizations. We examined four relevant Brazilian agile software development organizations, which undertake inter-team knowledge sharing according to their contexts and got feedback from an expert in agile methods implementation.

The criteria for selecting Brazilian organizations to participate in this qualitative study are the following: (1) the organization should have more than one agile software development team to allow for exploring cross-team knowledge sharing; (2) agile method(s) should be adopted for at least one year to benefit from intra-team culture of knowledge sharing; (3) the organization should apply practices for knowledge sharing toward organizational level (e.g., inter-team, inter-departments or organizational); (4) well-known and relevant organizations in marketplace.

Also, we include the convenience and availability of the organizations' context for answering the research question, the congenial relationship established between the organizations and the university/researchers, and the potential to learn from the research context. Yet, the participants are aware of the early stage of the research. Following is a contextual description of each case selected and its practices for the topic under study.

3.1. *Universo Online (UOL)*

Brief description. UOL is the major content portal in Brazil[e]. This large-sized company employs more than 1000 people. The expansion of services offered to the web market has demanded changes in organization's process as an attempt to deliver more software with speed.

The need to deliver in pieces and to think of maximizing value in each delivery has led to the Scrum implementation in the Research and Development (R&D) department in 2008. They first deployed Scrum in two pilot projects:(1) an evolution of an online marketplace software for buying and selling products, and (2) a new tool for managing complex sets of data and metadata, targeted to internal users [Maranzato et al., 2011]. Both projects were successful and proved that Scrum would fit to the company.

[e]http://sobreuol.noticias.uol.com.br/index_en.jhtm

However, a third pilot Scrum project aimed at rewriting a very large system and replacing an old and limited platform experienced problems to implement the method. They later learned that they were lost in pointless technical analyses, endless discussions, and personal conflicts.

After three years of Scrum implementation at the E-commerce area (a R&D department's area), they adapted it to another process, ITIL[f] (Information Technology Infrastructure Library), which is followed by the whole organization. Today, they consider mature in the method and throughout these years, they have also tried to implement, in a limited way, some XP practices, such as continuous integration, test-driven development and PP, when appropriate. However, they consider that there is still a long way to XP that depends on several organizational barriers, such as top management and developers' convincement, investment in infrastructure and training, and change development culture.

At the E-commerce area, where we accomplished this study, more than 75 people are employed to develop software for inter-related products in nine collocated Scrum teams, being six for the same product and three for different products.

Agile context. The basic ritual of Scrum is followed by the E-commerce area. Also, they complimented Scrum with some other meetings and rituals to suit their needs, for instance, they have defined three-week iterations that include planning, estimation, and close follow-up (daily and at the end of each iteration).

When they started to adopt Scrum, people from other areas, such as marketing or sales department, mostly assumed the Product Owner (PO) role. They realized the extreme importance of this role, which, in their dynamic context, demands a versatile project manager profile and a broad product vision, including technical, commercial and organizational issues. So far, emerging as a natural leadership, some scrum masters also act as PO's. In the core principles of Scrum, it could be considered wrong, but they achieved certain maturity in the process that allowed confidence in assuming this role.

Inter-team knowledge sharing approaches. They have adapted their Scrum of scrums meetings into other ceremonies that involve different teams from the same product. An experience report on the topic is presented in [Maranzato et al., 2011]. In these ceremonies they coordinate work and evaluate areas of integration and overlap. Also, everyone gets aware of the stories and raise questions or interdependencies. The impacts are discussed and when necessary, they schedule to detail specific issues about related stories.

"Mega Daily" is a meeting that occurs once every fifteen days. It involves all teams of the same product to discuss what they are working by focusing on areas of overlap and integration. Another ceremony is the "Mega Planning" that happens after all the teams' planning where they meet all the teams of the same product to present what each team will do in the next iteration to analyze integration, interference or inter-dependencies.

[f]http://www.itil-officialsite.com/

Finally, they have a review meeting to present what was developed in the iteration and one person of related teams is designated to attend these meetings.

They create knowledge redundancy with members' rotation among teams, especially when a team has problems in specific roles that were already overcome by another one. This 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 as they consider "one of the great drivers for documentation", generally needed to be disseminated, since everyone in the team has to create tests. So, the scrum masters agree to rotate members to help disseminate the role culture and when it is done, she/he returns to her/his original team.

Another initiative is rotate people to integrate an existing project or to compose a new team after the end of a project. As a member said, "We have done that a lot, kind of on purpose (...) we take a person from a team and put her/him in another one to encourage knowledge exchange". And another added, "We say that even if you're in a team now, this does not mean that you'll be forever part of this team. If needed, we will move people. So people end up knowing other parts of the application". Other member outlined, "I think it's important to make rotation of people between teams".

They also use implicit knowledge maps to implement special interest groups that are focused on solving problems, as a member said, "I have a problem to solve (...) or I do not know how to solve it, but there is someone that knows". Yet, when they have to develop a complex technical solution, they usually present the design of the solution to more experienced people from other teams to express an opinion about it or evaluate it. In more isolated teams, when they create a technical solution that is quite mature, they present it to everyone in the area.

They use a lot of the communication channels, for instance the mailing lists, to raise their problems to solve or share topics of interest. Informal seminars are also promoted when they attend important conferences to spread the knowledge apprehended in those technical events, however some say that this continues to work because "It is more than an exchange of knowledge, it is an accountability".

A similar initiative that did not work was the *TechDay*, a periodic technical presentation with the whole software development area, as a member said "Once I started the presentation and did not finish it, as nobody charged me I did not continue".

Another approach that has failed due to work overload was the E-Commerce area presentation, where each team presented to others about their application, but as an interviewee stated, "It lasted three meetings". A statement that depicts this context is "To study a new technology and make something with it to, you know, discuss. Sometimes, it is really cool to bring up new things, but sometimes we are so overloaded with things, then you have to do a presentation for two weeks later on a new technology. Then, it ends up just not working".

Wiki documentation is still a trouble for them, which is duplicated and without any standard, "Each one documents in a way, there is no standard to follow. We still do not know how to measure what is necessary to be on wiki". About what to document, "The

documentation that has no value to be maintained, it soon becomes outdated, obsolete. What is updated is what you have obligation to update to work".

Study groups worked only at the beginning, but then distorted. As a member outlined, "At the beginning, I thought it was very important to participate, because we exchanged knowledge among Scrum Masters, POs, etc., but it lasted shortly, just until reach the focus of addressing the original need, which was the Scrum implementation". They believe mostly on groups built-up focused on solving problems or specific interests or needs. Actually, they do not focus on establishing many formal meetings, even talking at the bar or in the elevator is relevant.

3.2. *Apontador*

Brief description. Apontador is a Brazilian medium-sized company that employs about 80 employees and is leader in local search in the Brazilian Internet. The company owns the leader local search platform in Brazil, with mobile applications being downloaded by more than 100.000 people. In 2010, Apontador was chosen as one of the top 250 global technology companies by the AlwaysOn[g]. Also, this company is among the 100 Brazilian most innovative companies by Information Week Brazil[h].

The open source software development is also valued by the company, thus its application programming interface (API) is open for developers to build their own applications using Apontador's platform and points of interest. This encourages sharing of projects and source code among internal and external developers. They undertake it as long as they do not compromise any core-business strategy.

By the time of this study Apontador was part of the LBS Local group that also included MapLink[i],a provider of routing and traffic service for Google Maps; ApontaOfertas[j], a daily deals aggregator; and Imobox[k], a real-state website. The group employs around 150 people altogether.

Agile context. In 2009, they started to adopt Scrum at the software development area of Apontador. In the beginning, it was only one cross-functional team, but after some changes in the company and the growth of the number of employees, they divided the team into three other fronts to attend three inter-related products. By the time of the study employed there, they had 13 members collocated in three teams with one Scrum master, also the manager. They also reduced the duration of their sprints for one week to get to meet new demands that are increasingly frequent.

The Scrum ceremonies happen with members of all three teams, as there are people that are shared among the teams (e.g. Scrum master, QA[l], web designers, and DBA[m]).

[g]http://www.aonetwork.com/AOStory/Announcing-2010-AlwaysOn-Global-250
[h]http://issuu.com/informationweekbrasil/docs/iwb_231
[i]http://maplink.com.br/
[j]http://www.apontaofertas.com.br/
[k]http://www.imobox.com.br/
[l] Quality assurance role.
[m] Database administrator role.

Not-shared team members have also stated in different occasions that they feel the need to know what other teams will be working on as they can share knowledge and experience and offer help in certain occasions. There is one developer that also operates in the 3 teams keeping a macro-view of the 3 inter-related products.

After deploying Scrum, they started adopting some XP practices after a training session in 2010 to improve their engineering, such as pair programming, continuous integration, user stories, small releases, metaphor, collective ownership, coding standards, simple design, refactoring, testing, 40-hour workweek, and on-site customer.

Inter-team knowledge sharing approaches. They freely employ face-to-face communication due to proximity, the agile methods cerimonies, and pair programming between members from different teams. These practices from Scrum and XP are especially applied to help creating knowledge redundancy, solving problems, and improving/reusing solutions.

As they claimed for more integration with other areas, they started to employ *Coding Dojos* by inviting all teams in the organization through their communication channels. It usually occurs after work not to affect liberation of employees and not to harm the daily routine of the areas. Coding Dojos occurred every fifteen days with people from different areas and different programming challenges. This practice promoted learning in the programming languages used in the organization.

Most audience appreciated the practice and in the retrospectives at the end, there were several statements praising the initiative that supported learning in the workplace. As a member said, “With the Dojos, different teams are learning about automated tests and programming best practices, which seemed to be impossible in the past”. Another aspect raised by the audience in retrospectives was the following: “It also creates a sense of unity in which all realize that are working in the same company, with common objectives and that the company is more than just 'our own team'”. Even occurring after work, the audience is low and sometimes they need to reschedule Coding Dojos especially in times of delivery or when people are working on task forces, as a member declared, “The pressure for delivery of other projects was high and we did not have the discipline to continue. For about one month ago, we had two sessions. I believe that now we are more aware that we must keep discipline and do the sessions even with only a few audience”.

They eventually promote internal seminars about specific topics and attend important conferences to help improving their solutions and work processes. They are planning to apply *Lightning Talks* to help identifying the knowledge owners, discussing solutions to problems and improvements, and developing insights within their areas and among other areas.

3.3. *Caelum Training and Innovation*

Brief description. Caelum is a medium-sized company that employs about 65 people, being 30 software developers, and is focused on learning, innovation and software

development. Training in the Java, Ruby, and Agile environments is the company's main revenue. Hence, learning and teaching is fundamental for the company.

Since its foundation, the company did not employ prescriptive methodologies to develop software. The founders, also software developers, were unsatisfied with the bureaucracy of the existing methodologies. Even without knowing they were applying several agile values and principles. The agile culture was hovering over their activities.

They also develop software for internal and external customers, and provide major contributions for the open source community. Contributions like VRaptor[n], Stella[o] and Restfulie[p] are acknowledged by the software development community. Those contributions work as a marketing strategy, increasing the company's reputation, and moreover, as a "catalyzer of learning and exchanging ideas with external developers" – citation from [Aniche and Silveira, 2011].

Due to its business domain, the company provided broad openness to observe the daily work of the developers and instructors. Apart from trainees, most of the instructors also work as developers, so they can acquire as much real life experience and excellence as possible to educate the students.

Agile context. In 2007, they started a course about Scrum, and later a course about XP. However, the XP course was shortly discontinued because of little industrial demand at that time. So, they began to apply XP in their software development through coaching by Mariana Bravo. Their actual focus is on XP practices such as PP, automated tests, continuous integration, simple architecture/design, and refactoring. XP practices like Test-Drive Development (TDD) [Beck, 2002] and small releases are not yet an imperative. They use a lot of automated tests and Kanban-driven software development with continuous flow delivery [Liker, 2004], not iterations. The other XP practices such as 40-hour workweek, on-site customer, collective ownership, and coding standards were already employed before deploying XP.

Then, in 2008, they added Scrum to their software development methods through coaching of Danilo Sato. In 2009, Caelum offered the official Certified Scrum Master (CSM) training by the first Brazilian Certified Scrum Trainer (CST), Alexandre Magno [Corbucci et al., 2011].

In early 2010, the Scrum course was reformulated to present agile methods in general, focusing mainly on the management practices. Later on the end of the same year, a new course was formulated in order to cover more technical practices such as unit and acceptance tests, test-driven development, and refactoring.

[n] First and most well known Java framework in Brazil, a local approach that has been even referenced as innovative by the members responsible for the Java EE specifications.

[o] An attempt to bring local issues from Brazil into programing code, such as zip code and Brazilian social security number handlers.

[p] Ahypermedia aware REST framework with international recognition.

Inter-team knowledge sharing approaches. They apply agile cerimonies, like standup and retrospectives, with all as a way to disseminate relevant experiences to all members. In the occasion of our research, retrospectives occurred periodically and when appropriate, however they perceived that it became delayed to happen and people used to forget on what to reflect. So, they started to schedule project and overall retrospectives in the white board. Another interesting concern was about improving their visibility by enriching their informative workspace (follow-up posters, Kanban board, tracking of velocity metrics and humorometer, etc.).

PP is part of their culture and they apply it in a very free way. It may occur with two, three or several people in a Superparing[q], and even in a distributed way with specific tools. A very interesting variation of PP stimulated by them is its use among different teams. They created this inter-team practice to encourage its adoption with as many people as possible to spread knowledge through the teams and to give more visibility of who is exploiting experiences in other projects. So, after pairing, they put a mark on a pair programming matrix (illustrated in[Aniche and Silveira, 2011]) to help evaluate the degree of knowledge sharing. Monthly they report the score and the winner(s) is(are) the one(s) that did PP with at least 70% of the software development area. The winner(s) is(are) rewarded with beers.

This broad rotation of pairs has improved knowledge propagation in their point of view, however in the occasion of our observations we did not witness any winner. Actually, they apply it in a free way for purposes of solving problems and common interest/needs.

They promote a technical lunch break known as “Brown Bag seminars” every fifteen days by addressing presentations of different topics with lunch afforded in part by the organization. At the end of the session, they make a retrospective to gather collective learning and raise feedback for improvements.

They also nurture other internal events. *TechDays* are employed every six months, where people prepare presentations on latter technological innovations. “Programming Sunday” occurs every two or three months, where pairs are formed to code several chosen projects (internal or open source). After every half an hour, the group does a small retrospective on what they did and pairs change. The company pays the lunch for all participants. An initiative that did not work was the “Technical Book Club”, because people considered the subject much static and often exhausted. So, they relieved the discussions to their mailing list, which are strongly used.

Coding Dojo sessions are also employed outside working hours. In the beginning they were just regular sessions, but they perceived that it becomes more effective when they establish more focused sessions considering purposes of leveling knowledge, solving problems, improving existing solutions, or developing new ideas in technologies.

They usually rotate people when composing new teams after the end of a project. The open space and their spatial rotation also help a lot on spreading knowledge at the

[q]Superpairing is a Coding Dojo considering the development of actual project stories.

workspace. When they need help, they just ask in the workspace and the help just comes up. Also, open source contributions and participations in formal and informal events, such as Conferences, blogs posts, forums, hang out at weekend for informal conversations out of the workspace. A further initiative they intend to cope is to apply Scrum for training preparation material. They have searched for providing more visibility about interactions among people and they still have to reevaluate some of their initiatives.

3.4. *ThoughtWorks (Brazilian branch office)*

Brief description. ThoughtWorks (TW) is a global Information Technology (IT) company[r] founded in 1993 and present in 8 countries, including Brazil, which employs about 1800 people. TW provides software development, consulting, and systems transformation services based on agile methods. Their main objective is not only delivering product, but also spreading the agile culture to other companies implementing ASD.

This company is pioneer in producing advanced and successful methods to develop software through several tools for managing agile software life cycle and best practices currently used in industry. As such, one of its employees is Martin Fowler, one of the authors of Agile Manifesto [Beck et al, 2001].

In 2009, TW opened a branch office in Porto Alegre (POA), RS, Brazil, which employs about 100 people (medium-sized company). We attended one lecture entitled "Theory in practice: developing software with Agile Methods at ThoughtWorks" given by two software development consultants[s]. In this occasion we could identify their endeavor to coordinate work in sub-teams and disseminate knowledge.

Since this branch office is located in POA, which is in a different state from researchers'address, it became impossible for the researchers, at that moment, to conduct observations. However, with their permission, we sent them a semi-structured questionnaire with open questions about the conditions, stimuli, and practices they use to motivate knowledge sharing among agile teams. This was an opportunistic case selection [Yin, 2008], however even with the limitations of the data collected, we could increase our view of the factors influencing the line of inquiry.

Agile context. As in the headquarters, TW-Brazil does not apply agile methods "by the book", they emphasize that they adapt XP, Scrum, and Lean Software Development, keeping agile values and principles in five large teams, one being composed of five sub-teams, each working on a project for the same client.

They work in pairs most of the day. Even varying from people to people, the time devoted to work in pairs, in their opinion, is an average of seven hours a day in most working tasks within the team (local) and with distant teams (remote). This includes pair

[r] http://thoughtworks.com/

[s] Luiza Pagliari and Marcos Valtas from TW-Brazil, along with Guilherme Silveira from Caelum, presented lectures at the "Agile Coffee" (http://www.thoughtworks.com/events/cafe-agil-em-sao-paulo) in June, 11th at University of São Paulo.

programming, writing e-mails, preparing internal or external presentations, book translation and even when they do not work. 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”.

Inter-team knowledge sharing approaches. They do not employ PP in teams of different customers, because the customer pays for the specific consultant hours, so they have to focus on the project. They use PP for creating redundancy of knowledge, and another interesting use of PP is for leveling knowledge with novices, as a consultant 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”.

Open spaces facilitate communication, “Open space spread information of all kinds without barriers, and many people with different roles sit together”. And disseminates knowledge among all teams: “If you have a conversation across the table and another one knows what it is, she/he enters the conversation, hears, becomes aware, and even sees the build breaking (…) In a few moments, everyone already knows more or less what your project is doing, what you just did. If the solution was cool, people schedule a ‘lunch and learn’ to present it at lunchtime”. ‘Lunch and learn’ is a practice for sharing knowledge with all teams’ members in which they present interesting topics in the organization during lunchtime. Due to the recent growth of the company, lunch is eventually afforded.

Open spaces help in identifying the knowledge owners for knowledge transfer: “This guy knows that system very much and this one doesn’t, so let’s put these two together. So we use a little of the knowledge matrix”. Another great use of open spaces for this endeavor is spatial rotation. Teams usually change their position in the room, “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. Then we switch teams to different places. So, this helps the team not to create a routine, and exchange other opinions, arguments, and discussions”. They also rotate people among teams, when needed.

Retrospectives with all employees are periodically employed to reflect on what to improve in the organization. Retrospectives of teams from the same customer are more frequent and they usually read the post-its of the “walling walls” prior to this meeting. In teams from the same customer, they also make daily standup meetings with all members: “As our team is getting too large, standup does not fit in fifteen minutes. So, the small teams within that large customer, make their own standup by raising ‘what I did yesterday’, ‘what I will do today’ and ‘what is interesting to say to the rest of the teams’. This happens quickly, just before the standup with all teams. And then, only one person from each team talks. It is more focused, less status to manager, but more challenges and new things that people have learned. (…) Then in the afternoon, as we work with teams from USA, we have smaller standup meetings with the teams via videoconference, with cameras and TV showing different rooms around the world. So, in our case we have two standups per day”.

They also feel responsible for spreading the agile culture in many agile events, as it is a common practice of other TW offices, especially head office. "As a way to disseminate the agile culture and make the whole world more agile, we like to give talks in conferences".

Something they believe that does not work at all for them is extensive documentation in wikis. They stated that document is 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 document only what is really necessary to avoid overhead and write good and readable code with automated tests.

The above approaches promote improvements, such as increase in redundancy of people to work in a story, decrease in individual dependence, greater readiness to fix bugs in different applications, quick insertion of new employees or transferred employees in projects (without having to spend days reading documentation).

3.5. *Massimus Consulting and Training*

Massimus[t] is focused on Agile Project Management training and coaching. Heitor Roriz Filho is a specialist in agile methods implementation and KM [Silva et al., 2010], working mostly as an agile coach and trainer. He has been dealing with ASD since 2004 and in addition to speaking for Agile/Scrum, Six Sigma and PMI conferences, Roriz is a passionate for Agile and Scrum. He is an agile champion [Corbucci et al., 2011] and truly believes Agile can change the way we work achieving excellence levels of precision and performance.

Given his experience in several software development organizations and also in KM, his point of view is important to triangulate data and to perceive if the results gathered from the organizations could be found in other organizations or if they were restricted to their context.

Regarding the practices, he reported, "teams exchange and generate knowledge on the ceremonies of the processes used (e.g., Scrum ceremonies), group dynamics, coaching sessions". He also added, "In terms of software design, best practices of agile software engineering: TDD, ATDD, Pair Programming, Continuous Integration, in short, XP".

When employing knowledge sharing practices in coaching consulting, he also needed to adapt some practices to share knowledge across agile teams, "I have used open space-based sessions[u] with very positive results (not pure open space sessions). BoF sessions leave things in the clouds, but they are very good to share, but not to generate results".

[t]http://www.massimus.com

[u]Open space session is based on the open space technology, an elaborate facilitation technique usually held on conferences [Dierkes et al., 2003], http://en.wikipedia.org/wiki/Open_Space_Technology, to provide really productive meetings, http://www.agileopen.net/on-open-space.

Roriz assigns the increased knowledge flow by these practices along with the whole organizational context, improvements such as product quality, technical excellence of the team, and software development process improvement.

4. Research method

Our aim was to explore factors and obtain richer and more informative results related to the inter-team knowledge sharing phenomenon in agile software organizations. For this reason, we chose a qualitative research method [Strauss, 1987] and established the research question guiding this inquiry as: "How inter-team knowledge sharing effectiveness may be achieved in agile software organizations?"

The method we considered most conducive to accomplish this study was grounded theory (GT) [Glaser and Strauss, 1967], because considering the social nature of knowledge sharing, which takes place in organizational contexts, it requires in-depth examinations to capture specific issues and human behavior from empirical data. Likewise, GT is a systematic method for discovering theory with a genuinely inductive approach, which is appropriate when there is no *a priori* theory for the research in hand.

The researcher starts from scratch without any theoretical model guiding data collection and analysis, but integrates existing theoretical models in the last phase of data analysis. The method offers detailed procedures for data analysis that allows for the emergence of original and rich findings rooted in the data.

To accomplish this qualitative study, we had to undertake it in four iterations of joint data collection and analysis. Through a process of theoretical sampling, we selected interesting aspects closely tied to the data to help us decide the analytical grounds to refine our next sample. This culminated in refinement of the research question, which is explained hereafter in Section 4.1. In this section, we present the iterations of joint data collection and analysis, and the limitations of the study.

4.1. *Joint data collection and analysis*

A continuous interplay between data collection and analysis is strongly recommended in GT to allow for the development of theory. Figure 1 depicts the iterative process of data collection and analysis in this study within and across the selected cases.

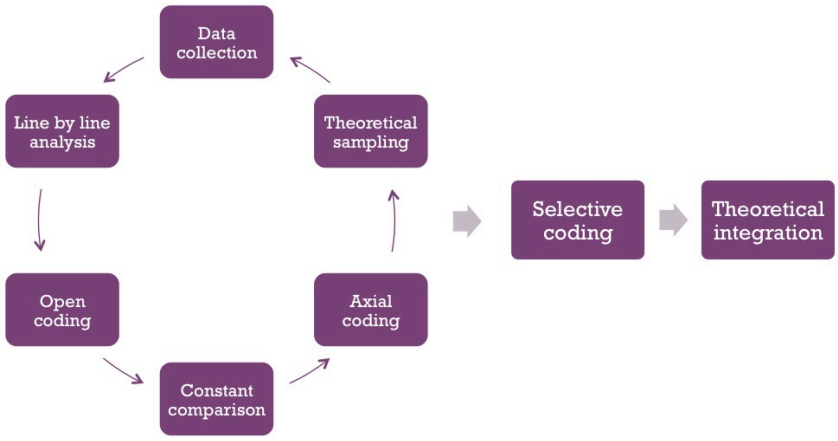

Figure 1. Cycle of joint data collection and analysis in the study.

We first began to collect data of each case. For each data collection phase, we developed a research protocol to guide the data collection process. This protocol helped us to define broad themes that represent the boundaries of the research question and build an initial guide for interviews and questionnaires.

We gathered data via semi-structured interviews with open questions, questionnaires, observations, and other sources such as lectures and papers as described in Tables 1 to 4. The interview guide of this study was developed before each data collection considering previous theoretical sampling, if any. Recorded interviews and lectures were transcribed. Observations and informal conversations were documented in field notes.

In observations we tried to be as discrete as possible to minimize threats to validity. However, we could perceive some of our influences in the field, for instance the participants' change in their work processes. The observations occurred from December 2010 to July 2011 and lasted from 15 minutes to 4 hours. We took text and numerical field notes and integrated our notes to produce detailed records (when necessary) from the interaction, collaboration, and knowledge sharing among teams. The quotes from the data gathered are presented later in this paper (Section 5) and were freely translated to English by the authors.

We analyzed data from transcribed interviews, field notes, detailed records, statements, dialogues, notes/transcriptions from lectures, and organizations' paper extractions (detailed in column "Others" from Tables 1 and 3). We analyzed raw data line by line to identify units of meaning and code them into *conceptual categories* (concepts) using the open coding technique. Data was highlighted, numbered to easily trace back to its source, copied, inserted into tables, partitioned and clustered in several ways.

The constant comparison method was employed with raw data and established categories. We worked back and forth between them, so new and more inclusive categories were developed. The so-called *notional categories* encompass relationships among the concepts. The selection of category names was based on the terminology used by the informants ("in vivo" codes) and on the terms brought by the researchers as a way

to make sense of the data (sensitizing terms). This allowed us to develop preliminary theoretical propositions and memos (research insights).

This cyclical process was applied in each selected case and relevant concepts and/or categories were identified to achieve *theoretical sampling*. This helped us to follow through on the next iteration with more sense of direction, subsequent questions, and listen and observe in more sensitive ways [Corbin and Strauss, 2007].

We closely examined the internal consistency of the categories and properties[v] 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.

We systematically compared across cases until no new categories or questioning of existing ones were generated toward reaching 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).

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

Considering the concept of "trustworthiness" for qualitative validity [Guba and Lincoln, 1980], we provided detailed description of the research context and the assumptions that were central to the research as a way to enhance *transferability*. We checked and rechecked the data throughout the study for developing theoretical sampling and accomplished peer debriefing (the co-authors discussed the findings with each other and with organizational members through feedback meetings/presentations and reports) to strengthen *credibility*. *Dependability* was enhanced by rigorous theoretical sampling and by presentation of our judgments about potential for bias or distortion. And we used multiple sources of evidence for theoretical sampling to increase *confirmability*. We also share that our own ontological and epistemological positions impacted on our use of GT.

As a result of the research we developed an emergent theoretical model. This means theory for explaining [Gregor, 2006], since we explained primarily "what" are the factors influencing the phenomenon under study, "how" and "why" they occur in the context ("when" and "where").

Figure 2 depicts the four iterations of joint data collection and analysis, along with the description of iteration's period and organizations participating. In the following sections, we provide detailed information on each research iteration.

[v] Characteristics or components of an object, event, or action.

Figure 2 – Description of iterations, period and participants.

4.1.1. *Iteration one*

We started our study in November 2010 with a very broad research question, which was focused on understanding how agile software organizations undertake organizational learning. The first data collection was structured around three broad themes:(1) the characteristics of agile method(s) adoption towards organizational learning, (2) the motivation and barriers to share knowledge in wider organizational levels, and (3) the approaches applied for generating organizational knowledge. Table 1 presents the participating organizations and the gathered sources of evidence in this iteration, which resulted in a total of 76 transcribed pages.

Table 1. Sources of evidence of the data collected in the 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., 2011a], 6 pages.
Massimus Consulting	1st semi-structured interview: specialist (1). The interview resulted in 15 pages.	N/A[w]	Document: Paper #2 about the analysis of 'Ba' during Scrum process [Silva et al., 2010] in Portuguese, 15 pages.

The open and axial coding process was done in all iterations similarly. In this iteration, it was made separately for evidences of UOL and of the specialist, and then we compared both analysis to generate concepts and possible categories. We identified the need to triangulate with another organization to help in identifying and enriching the related issues to generate organizational knowledge in agile software development teams.

[w] Not applicable.

4.1.2. *Iteration two*

In this iteration, we included Apontador along with UOL, as presented in Table 2, to explore the applied approaches in different organizational contexts and understand the issues influencing the generation of organizational knowledge. The gathered sources of evidence in this iteration, which resulted in a total of 31 transcribed pages.

Even applying approaches for knowledge sharing, many of the informants explained that they had difficulty in undertaking it in wider levels and applying it effectively. They say that the knowledge sharing approaches are apart from the work processes and justify by saying they have to focus on delivery of value to customers.

Table 2. Sources of evidence of the data collected in the 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

4.1.3. *Iteration three*

Then, after data analysis of the previous iteration, our theoretical sampling for the third data collection was focused on exploring the factors influencing the occurrence and effectiveness of inter-team knowledge sharing in agile software development organizations.

In this iteration we included two more organizations, Caelum and ThoughtWorks, out of four organizations to observe and/or interview, along with another interview with the specialist, as presented in Table 3. We interviewed and observed new respondents/informants, as well as second-round interviewees, resulting in a total of 143 transcribed pages.

This iteration occurred from June to August 2011 and enriched the study, since we were more mature within the thematic area and we gathered data from multiple organizations within different contexts. The data analysis was made within and between the cases, and allowed further development and understanding of the categories, properties and their relationships. The interaction ended when reassessments generated no new categories, sub-categories, or questioning of existing ones, resulting in the

Table 3. Sources of evidence of the data 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.	
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.	
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[x]: 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	

building of a theoretical construct on issues influencing the occurrence and effectiveness of inter-team knowledge sharing in agile software development organizations.

[x]In June 11th, 2011 we attended a lecture of both professionals, entitled "Theory in practice: developing software with agile methodologies at ThoughtWorks". They presented the company's history since its

4.1.4. *Iteration four*

After the third round, we spent September and October 2011 developing a systematic refinement and integration of previously generated notional categories to define a matrix of dimensional themes. After, we went to the final coding step. Then, in November 2011, we presented our findings to the organizations in feedback reports or in feedback meetings/presentations, when possible. We tried to involve as many people as possible with different roles to evaluate the results. This improved qualitative validity, as the participants could judge the results. Any discordance was adjusted until reach consensus.

We also tried to confirm our perception on what does it mean effectiveness of a practice applied for inter-team knowledge sharing in the perspective of the agile team members of the organizations. So, new interviews were made with different respondents as presented in Table 4. In this stage, we checked if any data was missing or invalid and ensured that the collected data was sufficient to answer the research question.

Table 4. Sources of evidence of the data collected in the fourth cycle.

Organization	Interview	Others
UOL	Semi-structured questionnaire: Executive, PO, Scrum masters and developers.	Feedback report and presentation: Executive (1), PO (2), Scrum master(1) and developer (1).
Apontador	Semi-structured questionnaire: Scrum master and developers.	Feedback report and presentation: Scrum master (1), developers (8) and QA/Tester (1).
Caelum	Semi-structured questionnaire: software development consultants/trainers (9).	Feedback presentation: software development consultants/trainers (16).
ThoughtWorks (Brazil)	Semi-structured questionnaire: software development consultants.	Feedback presentation: software development consultant (1).

4.2. *Limitations of the study*

We had some limited experience in conducting this empirical study, because in three organizations it was not possible to observe in an extensive way the daily work of practitioners, first because of distance (another Brazilian state) and second because of confidentiality matters and researchers influence in practitioners work. In both 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. Only Caelum, due to its

foundation in United States, their acknowledged professionals in software engineering (like Martin Fowler) and agile methods communities, their business focus, their way of working, agile methods adapted to suit their business goals and their relationship of trust with customers. In this occasion, we could talk to them about their organizational context and their approaches for learning and knowledge sharing across five large teams.

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 research topic, 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 manage to avoid that our knowledge of the field did not lead us to preconceived theoretical ideas that could hinder the emergence of ideas firmly linked to the data.

A well-known difficulty in GT is to derive all the detailed materials into a journal paper. For this reason, in the following sections, we distill the results of the third axial coding, the conditional matrix of notional categories and dimensional themes, the selective coding core category and the narrative, theoretical model, and theoretical integration.

5. Results from grounded theory approach

After the third wave of iterations we collapsed the resulting categories into 11 more encompassing notional categories presented in Tables 5 and 6, along with examples of evidences. This was done by Strauss and Corbin's paradigm model for linking previously established categories [Corbin and Strauss, 2007]. We exercised the relationships among categories, then consulted instances of the categories in the raw data and constructed a logic of association among them before deciding whether to join them or not. This approach identifies related phenomena, causal conditions, contexts, facilitating or restraining conditions, action/interactional strategies, and consequences of actions.

Table 5. Categories resulting from the third wave of data analysis (Part 1).

Notional category	Conceptual category	Example
1- Strategy	Mission and Vision statements	"In the Vision statement, there is explicitly aim to share knowledge and experiences for innovation and cohesiveness."
	Goals and objectives	"We needed to be organized by product to increase focus, get the whole picture and develop quickly"
	Work processes	"Exchange good practices is a challenge, since our major focus is on the product (…) You forget the other teams that work with different products" Field note #1.37: They use to consider knowledge sharing as extra work, not as a resource for accomplishing long-term innovation and improvement.
	Feasibility concerns	"I think it's feasible, however we just have to be careful with the dosage"
	Resource management	"Learning has to have a low cost. To have a low cost, it has to be natural" "Every time we had *Techday*[y] (...) it is like a holiday to encourage people to go"
2- Structure	Physical	"We work in open spaces with few partitions and walls"
	Hierarchical	Field note #3.47: The directors are close to management and operational members, and management sits along with operational.
3- Culture	Shared beliefs	"As a way to disseminate the agile culture and make the whole world more agile, we like to give talks in conferences"
	Freedom of expression	"There is less imposition by managers, and more freedom of expression"
	Values and attitudes	"Developers have a greater commitment to the product"
	Patterns of behavior	Field note #3.109: Employees organize their own time for studying or searching for new technologies; for engaging in master/doctorate courses and academic researches; for preparing the training classes and material; for developing innovative products for clients; working on open source projects; refactoring code, etc.
	Strengthening of relationships	"The relationship is more transparent"
	Particular language development	"That is a sprint thing!"
4- Environment	Business domain characteristics	"The company is undergoing a phase of growth accompanied by changes."
	Tension between change and stability	"The pressure for delivery of other projects was high and we did not have the discipline to continue [The respondent was talking about the initiative of doing Coding Dojo sessions more often]".

[y] All-day internal eventfor sharing knowledge about technology and best practices for software development.

Table 6. Categories resulting from the third wave of data analysis (Part 2).

Notional category	Conceptual category	Example
5- Top management support		"Top management is more centralized and does not give much freedom to experiment."
		Field note #3.68: The high support for improving expertise and delivering high quality services and products comes firstly from the founders. They were precursory in freeing up employees' working time, and in contributing to open source projects. For instance, they have established one day per month to stop work and refactor code or contribute to open source projects (for the ones that work for external client's project).
6-Leadership characteristics	Intention to share knowledge	Field note #2.44: The Scrum master has supported the occurrence of several approaches, for instance we observed the Coding Dojo sessions and their striving to encourage its frequency and wide attendance of members from all organization.
	Encouragement to knowledge sharing	"So we support, encourage, but when it's time to put it into practice, we just prioritize other things."
	Natural leadership	"They have acquired the leader role by recognition of others"
	Assignment	"Leadership helps in avoiding instability and in guiding to the objectives."
	Team autonomy	Note from lecture #4.23: Leadership operates as coach to empower self-organizing teams.
7- Communication	Flow	Field note #3.19: Due to mutual interplay of physical structure and culture, communication flow is highly facilitated, empowered and it occurs without obstacles.
	Channels	Field note #3.87: They use a lot of the communication channels, e.g., mailing lists help raise problems or share topics of interest. "Speech and drawings (sketches)"
8- Integration among teams and projects		Field note #3.124: Due their great openness to communication and cohesiveness, there is high integration among teams, even without strict needs for projects integration and overlap.
9- Agile method(s) adoption	Number of agile methods	"We do not apply agile 'by the book', we adapt XP, Scrum and Lean"
	Time of implementation	"We were born agile"
	Level of adoption	"The working process is well established across the area"
	Level of maturity	"We reached a level of maturity in Scrum"
10- Stimuli	Problem-situations	"It is related to the project, a problem"
	Common interests/needs	"It's the need; we help each other because of a common goal. If we did not have the necessity, they would be working there and we here separate."
	Sustainable pace	Field note #3.98: They are quite mature in establishing a sustainable pace for achieving their goals linked to the organizational ones.
	Incentives	Field note #3.14: Lunch is afforded to engage "Brown Bag seminars"[z], "Programming Sunday", *Coding Dojo* sessions.
11- Inter-team knowledge sharing	Adapted approaches	Paper #3 extraction: "The team proposed a Pair Programming Matrix on the wall" Field note #3.7: They employ PP between members of different teams and measure in a PP matrix.
	Purposes	"This guy knows that system very much and this one doesn't, so let's put these two together. So we use the knowledge matrix."
	Effectiveness	"*'Oh, let's create an interest group in QA'*, this lasts two, three meetings. If the interest is not very strong, it just dies."

[z]http://en.wikipedia.org/wiki/Brown_bag_seminar

Through intensive comparison, we linked the notional categories at two dimensional levels, commonalities and differences. Table 7 presents a conditional matrix of relationships among notional categories and corresponding dimensions.

Table 7. Conditional matrix of notional categories and dimensional themes.

Notional categories	**Dimensions**	
	Commonalities	**Differences**
Strategy	Fast delivery of value to customer.	Strategic alignment and knowledge sharing as part of the work processes.
Structure	Teams' closeness.	Autonomy and self-organization, hierarchical formalities, and physical barriers.
Culture	Members' engagement.	Organizational culture as boundaries to values and attitudes toward knowledge.
Environment	Difficulty in balancing the tension between stability and change to map the environment.	Creation of a momentum to learning.
Top management support	Consider knowledge as important resource to long-term strategies.	Degree of self-organizing and autonomous teams to embrace organizational changes.
Leadership characteristics	Empowerment of teams.	Hierarchical leadership versus leadership by acknowledgement. Degree of self-organizing and autonomous teams to embrace organizational changes.
Communication flow and channels	Communication enriched.	Level of osmotic communication.
Integration among teams and projects	Integration depends on common purposes.	Openness to establish integration.
Agile adoption	Formalization of agile culture	Number of agile methods adopted, level of adoption, time of implementation and level of maturity.
Stimuli	Problems, common interests/needs and incentives.	Organizational context as boundaries for the definition of sustainable pace.
Inter-team knowledge sharing	- Agile teams know the agile practices value. - Failure in employing codification strategy (e.g., wikis, documents, etc.) - Focus on practices for socialization. - Adaptation of practices to be more agile and flexible.	- Organizational posture to promote practices' effectiveness. - Number of employed practices, its frequency, its level of formalization and reassessment in the organization.

Although the organizations under study understand the importance of considering sharing and learning to improve their work, we identified differences on how they cope with that. Some of them are very concerned about applying it effectively without affecting their agility in deliveries, and others create an organizational posture toward knowledge.

Inter-team knowledge sharing is somehow employed in all organizations. A relevant concern most interviewees declared is that knowledge is not properly reused, they usually "reinvent the wheel" or redo work, instead of saving time by reusing existing solutions from other teams. Because of that, as they can they implement knowledge sharing even after working time.

We also identified that organizations with clear posture toward knowledge employ more practices for sharing knowledge, which are more frequent and become naturally part of their work processes. For instance, at Caelum, when we started observations there, they used to promote a technical lunch break known as "Brown Bag seminars" every fifteen days. At the end of the fourth research iteration, they told us they were applying it weekly. Yet, they have abandoned the pair programming matrix and started to adopt other new approaches, such as pair programming with external teams, projects marathon and goals board. They also replaced the daily meeting with all for a weekly meeting (called *Dev Weekly*). Also Roriz claimed he needed to adapt some practices, for instance Open Space-based sessions, to share knowledge across agile teams, as presented in Section 3.2.5.

After linking the notional categories, we went a little further and collapsed them into more inclusive categories: conditions, stimuli, and inter-team knowledge sharing practices, and effectiveness. The narrative around them, often termed as integrative memo, is described as follows and is illustrated in Figure 3. The **inter-team knowledge sharing effectiveness** in agile software organizations may be achieved by the application of **practices for socializing knowledge**. They are grounded by the organizational **conditions** and **stimuli**, which operate as influencing factors that empower or hinder the process of inter-team knowledge sharing. These practices have proper **description** and classification by specific **purposes**. **Effectiveness** is evaluated by the **level of purpose achievement**[aa], **frequency**, **level of formalization**[bb] and **reassessment** of the practices in the organization.

Inter-team knowledge sharing practices. The knowledge sharing initiatives employed by the organizations under study are regarded as being a socialization process. This means applying purposeful practices with people-focused and lower-cost strategies, crucial for emphasizing interactions and dealing with tacit knowledge.

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 they focus more on practices for socializing knowledge by adapting existing practices to suit their needs and to be more agile and flexible.

The main purposes raised by the organizations are to:

- Identify the knowledge owners: The interviewees state the need to facilitate finding people knowledgeable in a specific subject in order to contact them in the organization. Due to the complex nature of SE and the encouragement of intense direct verbal communication in ASD, human-centric strategies for a quickly people identification are fundamental. As a team leader at UOL stated, "I have a

[aa] The proposed levels are: (1) fully meets; (2) partially meets; and (3) does not meet.

[bb] The proposed levels are: (1) the practice is still little known and informally adopted; (2) the practice is widespread and accepted, intended to formally adoption; (3) the practice is widely accepted and formally adopted; (4) there is no intention to formalize the practice, since it should become a natural adoption; and (5) the practice is naturally adopted (widely acknowledged by the organization, but it was not formalized, it just became a natural adoption); and (6) the practice is not adopted anymore.

problem to solve here in my project", then "Oh, I do not know how to solve it, but there is someone who knows".

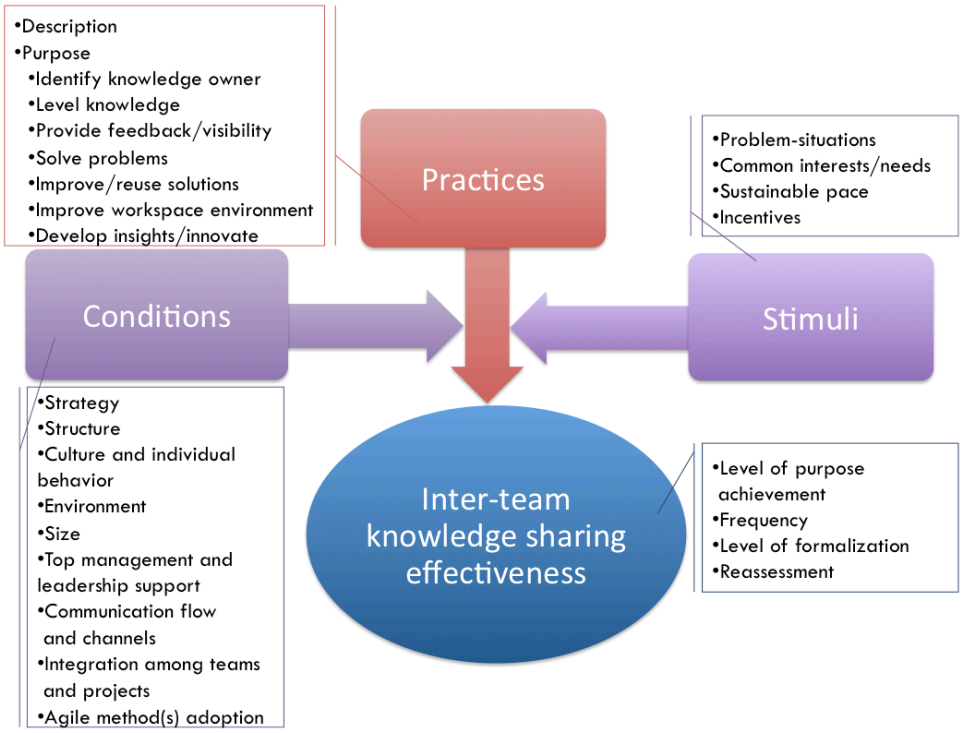

Figure 3 – Theoretical model for inter-team knowledge sharing effectiveness in ASD.

- Level knowledge: Purpose mainly referred to promote knowledge redundancy, technical excellence and larger 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".
- Provide feedback/visibility: Improve inter-team feedback flow is also considered important to scale learning through established systems, processes, practices and routines of the organization. Also supports to establish challenges and the outcome expected from professionals and respond to changes. Visibility is especially employed in projects with areas of overlap and integration. However, it is also employed to bring wider projects' view to other levels of the organization and raise everyone's awareness of what is happening in other projects. As a developer at Caelum stated, "It is important to have a broader vision of 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".

- Solve problems: Encourage inter-team interaction by taking joint actions to solve problems. Problems raised by teams mobilize people in the organization to share knowledge and rethink or restructure their assumptions and actions. As a scrum master of six teams at UOL posited, "The need to solve a problem drive to lot the knowledge exchange. We help each other to achieve a common goal. Without it, teams would be working isolated".
- Improve or reuse solutions: The motivation is to avoid "reinventing the wheel", because teams benefit from saving time by reusing or improving an existent solution. 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 better addressing the problem".
- Improve workspace environment: According to their point of view, knowledge sharing also improves professionals' interactions and strengthens their relationships.
- Develop insights or innovate: The participants in the study acknowledge inter-team interactions as a promoter of insights refining and development. For instance, members of organizations Apontador and Caelum report they are encouraged to use the infrastructure to develop new ideas for the open source community. Organizations more focused on creating innovative products or services are concerned about investing on this purpose. A development manager at organization Caelum stated they employ several approaches to innovation as a way to accomplish the strategic directions, "We need to study new things in order to be able to teach it when gets 'fashionable' in the marketplace (...) so with our research, our study, we have to be ahead".

Conditions. The main influencing conditions raised by the study consist of 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.

Some of the organizations explain that they do not have time to share knowledge because their business domain require quick and frequent delivery of working software. Roriz (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".

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 that the team gets mature in terms of project vision and also making team members understand that the organizational gears, which are eventually unfamiliar to the project's reality, are essential to the continuity of the organization".

A member at UOL also reported the relevance of an adequate structure for wide agile method adoption, "We saw a company that is structured for adopting XP, the office was

set up in order to meet the process and is very different (…) the layout needs to be adapted so that two people work on the same machine".

Agile software development teams achieve great levels of commitment, transparency and responsibility, 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.

To change organizational values and principles, it is necessary a complex endeavor, as Roriz said, "Change value is too complicated (…) It can be accomplished in two ways: by love or by pain (…) few trainers or coaches know how to make this change 'by love', trying to reach the root of the problem". By his experience, he posits that top management is usually resistant to embrace wide changes. As he said, "They deceive themselves by implementing the processes, but without changing the values (...) they can go through thinking that they are implementing Scrum, but they're not".

Learning requires both change and stability [Fiol and Lyles, 1985]. We noticed that the organizations had to deal with this environment tension 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.

We identified that agile adoption in the organizations, such as the number of agile methods adopted, level of adoption in the organization (e.g., team, area, organization, etc.), time of implementation and level of maturity (beginner, intermediate, advanced) impacted on the organizations as they became more prone to embrace changes and learn. Roriz claims that the agile methods impact on communication flow and KM processes because "agile methods promote interaction of various forms and the team is involved in everything". Leadership characteristics are strongly related to the characteristics of the adopted agile method(s).

An intricate relationship of strategy, structure, culture, environment and top management support was identified as one influences the others and vice-versa. Strategic directions and environment impact on culture, structure, and top management 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.

Stimuli. In the study, the main motivation for learning and sharing knowledge are problems, common goals/interests, and incentives. Roriz 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 knowledge sharing strategies.

Otherwise, organizations focused on long-term endeavors and major posture toward knowledge, apply more proactive strategies. The aspect of sustainable pace is something still over discussion in the organizations, but we noticed it is related to organizational posture toward knowledge. Some have established a "very maddening pace of delivery, delivery, delivery" (statement of a respondent at UOL), as time is not fully considered as a resource for learning and sharing knowledge. Some have to accomplish cross-team knowledge sharing practices after work not to affect liberation of employees and not to harm the daily routine of the areas. For instance, an interviewee at Apontador said, "We have difficulty in defining a sustainable pace in which knowledge sharing is a part of the activities during the sprint".

The selective coding is the final coding step in GT, which is focused on identifying the core category(ies) and in explicating the story line of the phenomenon under study. It is difficult to argue that one notional category is more central to the understanding, so our selective coding resulted in a new main category: **Context toward inter-team knowledge sharing**. And the story line is as follows, "The more knowledge-driven are the organizational conditions and stimuli, the more the practices for knowledge sharing across agile teams are created, sustained, adapted, and even abandoned spontaneously, hence effective for the organization".

6. Discussion

Based on evidence from four organizations and triangulation of data from a specialist in agile methods implementation and KM processes, we recall our research question "How inter-team knowledge sharing effectiveness may be achieved in agile software organizations?" and start the discussion by the following comparison, as the effectiveness of the agile practices rely on the application of agile values and principles [Dingsøyr et al., 2010], likely the inter-team knowledge sharing effectiveness in agile software development rely on the acknowledgement of an enabling context [Von Krogh et al., 2000]. Knowledge creation is a fragile process, often subject to strong barriers, hence depending on the context, some practices may work and be institutionalized and others may not.

Hence, the breadth of knowledge sharing among levels of learning (individual, team, inter-team and organizational) depends mostly on the organizational conditions in which it takes place and on the stimulus for leveraging interaction, along with the approaches for enhancing learning and sharing [Nonaka and Takeuchi, 1995] [Nonaka, 1988]. [Von Krogh et al., 2000] state that it depends on the energy and sustained commitment an organization puts into knowledge creation. Also, recent studies acknowledge the enabling context as crucial for the knowledge sharing process [Choo and Alvarenga Neto, 2010] [Wang and Noe, 2010].

We made a comprehensive literature review seeking for influencing factors for knowledge sharing to provide the last step in GT that is theoretical integration. Several seminal authors posit that a proper organizational context positively influences the knowledge sharing initiatives [Easterby-Smith and Lyles, 2011] [Dierkes et al., 2003]

[Von Krogh et al., 2000] [Nonaka and Takeuchi, 1995] [Drucker, 1995] [Mintzberg and McHugh, 1985] [Fiol and Lyles, 1985]. Table 8 provides a list of influencing factors.

Table 8. Factors influencing knowledge sharing from literature review.

Author(s)	Factors
Fiol and Lyles (1985)	Culture, strategy, environment and structure.
Nonaka and Takeuchi (1995)	Intention, autonomy, fluctuation and creative chaos, redundancy, and requisite variety.
Drucker (1995)	Strategy, horizontal structure, and culture.
Davenport and Prusak (1998)	Strategy, horizontal hierarchy, culture, type of valued knowledge, mutual trust, and tolerance for admitting mistakes.
Von Krogh et al. (2000)	Knowledge vision, right context, knowledge activists, mutual trust, active empathy, access for help, leniency in judgment, and courage.
Dierkes et al. (2003)	Management support, reward system, structure, culture, leadership characteristics, knowledge sharing context, communication channels, type of valued knowledge, doubt, and sense of safety.
Lin (2008)	Structure, culture, and trust and commitment.
Chang and Thong (2009)	Top management support, subjective norm, and organizational culture.
Wang and Noe (2010)	Management support, rewards/incentives, organizational structure, culture/climate, leadership characteristics, sharing context (online, face-to-face, etc.), and goals and values.
Joia and Lemos (2010)	Common language, mutual trust, relationship network, reward, hierarchy, favorable environment for questioning, and media.
Chiri and Klobas (2010)	Commitment, trust, and learning orientation
Choo and Alvarenga (2010)	Strategy/Structure, social/behavioral, cognitive/epistemic, information systems/management.
Donate and Guadamillas (2011)	Knowledge-centered culture and knowledge-oriented leadership.
Highsmith (2011)	Adaptive leadership, envision-explore culture, flexible strategy, purpose alignment.

Deep organizational changes should be considered for enabling knowledge creation, as well as the continuous process of questioning and reconsideration of existing assumptions to respond to changes by making incremental adjustments or organizational transformations. [Kotter, 2002] argue that embrace changes is a good source of feedback in organizations and the reason why people change relies mostly on the dynamic process "see-feel-change". The effect of emotion is also a concern in organizational learning theories, since emotions are an integral and inseparable part of organizational life informing culture and strategy, often functional for the adaptation to the environment [Easterby-Smith and Lyles, 2011]. Considering the relevance of adaptive capacity, we propose that:

P1. The greater the adaptive capacity of the organizational environment, the greater the inter-team knowledge sharing effectiveness.

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 space in which knowledge is shared, created, and utilized, since knowledge needs a context in order to exist. The main objective of *ba* is interaction and it needs to be 'energized' (stimulated) to become active and build meaning into workspace.

[Takeuchi and Nonaka, 2004] argue that enabling knowledge creation has a close connection to organizational structure and strategy that affect all other influencing factors, for instance top management commitment to initiatives of knowledge creation. Also [Von Krogh et al., 2000] report that open spaces reflect the new logic of knowledge intensive organizations, facilitating the continuous exposition to all kinds of operations of the organization. They also posit the need for mobilizing knowledge activists. Middle managers could assume this role, but the authors state that any member can assume it. Because of that, we also propose that:

P2. The greater the consideration of knowledge as a resource in organizational strategies, the greater the inter-team knowledge sharing effectiveness.

P3. The greater the flexibility in organizational structure with few hierarchical levels and physical barriers, the greater the inter-team knowledge sharing effectiveness.

P4. The lower the organization size, the greater the inter-team knowledge sharing effectiveness.

P5. The greater the top management and leadership support, the greater the inter-team knowledge sharing effectiveness.

Organizational culture is another crucial factor because it guides the organizational functioning and members thinking and behavior. Yet, as [Pettigrew, 1979] state, it is fundamental for putting the social structure of an organization together and for managing organizational change and renewal. In this study, organizational culture and the individual behavior toward knowledge sharing are mutually related. [Joia and Lemos, 2010] explain this relation by reporting that both require a favorable environment for questioning, relationships of trust between the individuals, development of a common language, and internal communication flow. For this reason, we propose that:

P6. The more the knowledge-centered culture and individual behavior, the greater the inter-team knowledge sharing effectiveness.

P7. The greater the communication flow and extensive use of communication channels, the greater the inter-team knowledge sharing effectiveness.

In theory, rewards and incentives, such as career consequences, bonus, gifts, are generally considered the keys for motivating knowledge sharing [Chiri and Klobas, 2010] [Wang and Noe, 2010] [Chang and Thong, 2009] [Stewart, 2003] [Davenport and Prusak, 1998]. In our study, we observed that implementing collective incentives, instead of individual incentives, greatly benefit the inter-team knowledge sharing process. That is why we propose that:

P8. The greater the collective incentives, the greater the inter-team knowledge sharing effectiveness.

In a recent study, [Pink, 2011] posits that incentives do not motivate knowledge workers. For the author, other aspects motivate them, which are autonomy, mastery, and purpose. In ASD, autonomy is accomplished by self-organizing teams, whose team members have a degree of autonomy over how they work and how decisions are made [Highsmith, 2011]. And mastery is enhanced by the steady chase for technical excellence and improvements in software design, which reflect on greater agility for the project [Cockburn, 2006]. We observed that this quest for continuous improvement has affected the inter-team knowledge sharing process, especially in experienced agile teams. That is why we propose that:

P9. The greater the experience in agile methods, the greater the inter-team knowledge sharing effectiveness.

[Pink, 2011] also reports that inspiring aspects (purposes), instead of extrinsic motivators, are the greatest drivers that foster long-term transformation process for knowledge workers. We also noticed in our study that most agile teams apply knowledge sharing initiatives when they face teams/projects integration and purposes, such as to solve problem-situations and to achieve common goals or interests. Then, we propose that:

P10. The greater the integration among teams and projects, the greater the inter-team knowledge sharing effectiveness.

P11. The greater the stimuli for solving of problem-situations, the greater the inter-team knowledge sharing effectiveness.

P12. The greater the stimuli for accomplishing common goals or interests, the greater the inter-team knowledge sharing effectiveness.

[Davenport and Prusak, 1998], as advocates of organizational hierarchy reduction, state that status and rewards should go to knowledge owners and the time and meeting places are essential to encourage knowledge sharing, along with organizational conditions. As well, [Terra, 2001] highlights time as an indispensable resource for knowledge creation. In our study, few organizations consider time as a resource for knowledge sharing, the reason they point out is the difficulty in establishing a sustainable pace between delivering value quickly to customer and accomplishing long-term strategies.

[Highsmith, 1999] in his seminal work that is behind the agile movement, declare that software development is context-dependent. Organizations have to become adaptive and it requires profound cultural shift. This adaptive concept is also considered in recent work [Highsmith, 2011], by emphasizing adaptive leadership to accomplish enterprise agility -

the next horizon for agile software development. This means extending agility from basic software delivery to continuous delivery through the transformation of businesses to deliver complete solutions early and often. The author posits that to accomplish long-term strategy, the organization should create a sustainable pace considering "Do Less" and use the time and money saved for reducing technical debt, for new innovation, and for improvement initiatives, which are in line with the implications of KM initiatives. For that reason, we propose that:

P13. The greater the establishment of a sustainable pace, the greater the inter-team knowledge sharing effectiveness.

P14. The more the agile software organizations have an enabling context toward knowledge, the more they are encouraged to share knowledge through proactive initiatives.

In the encompassing explanation of the spiral of organizational knowledge creation [Nonaka and Takeuchi, 1995], knowledge is not seen as a stock, but as a continuous flow in ontological and epistemological dimensions to generate organizational knowledge through the levels of learning. Even if employees leave the organization, knowledge remains because of the alive and continuous dynamic.

Another relevant theory to articulate is social network, because it is an approach that may reveal and provide awareness of the flow of knowledge among actors to improve knowledge sharing opportunities [Haythornthwaite, 1996]. [Wang and Noe, 2010] also recognize that social network theory provides another perspective on understanding knowledge sharing, since employees do not work, learn, or share knowledge in isolation, but embedded in social networks. The authors also relate this theory to communities of practice by reporting that when a formal or informal group is formed, its members bring with them not only their own knowledge, skills, and abilities, but also their social connections. The seminal work on communities of practice by [Brown & Duguid, 1991] indicate that the way people actually work differ from the organizational work descriptions, and significant learning and innovation is generated from informal communities of practice in which they work.

Lastly, it is important to remind that knowledge is socially constructed [Berger and Luckmann, 1966] and as [Easterby-Smith and Lyles, 2011] state, socialization and learning are inseparable processes. Since software developers have difficulty in getting aware of their knowledge, so a good strategy is to socialize knowledge and notice the emergence of organizational knowledge as a consequence of this process.

7. Implications for practitioners

Our research denotes implications for practitioners. First, agile software organizations need to understand the importance of scaling the tacit knowledge within team members. Furthermore, understand the importance of inter-team knowledge sharing process for the organization continuity and turn it as part of the organization work processes. The

organizations under study acknowledge realized benefits of improved technical excellence, software quality, communication, collaboration, incremental adjustments, and organization transformation.

Second, this process is people-oriented. Thus, to succeed and facilitate cross-team interactions, individual, group and organizational barriers should be reduced in order to establish an enabling context that will ground the employment of purposeful knowledge sharing practices. With the right context, the practices are naturally adopted, adapted, sustained, and even abandoned.

Third, it implies in low-cost strategies, since it does not require knowledge management department or executives, but the mobilization of professionals as knowledge activists, who can be concerned of energizing and coordinating knowledge sharing efforts linked to action. The strategies should consider the existing organizational conditions and stimuli to initially improve them, if possible. Then, when all or some relevant conditions and stimuli are overcome, apply practices according to the purposes relevant for the organization.

Lastly, we identified that most organizations under study underutilize documentation, because of the context-dependent nature of software development (for instance, people are not aware of what is fundamental to register in a wiki), the agile methods focus on avoiding extensive documentation, and the lack of standard on what to and how to document. However, several inter-team knowledge sharing practices may help on externalizing tacit knowledge and on emerging solutions for these issues and improve the production of a documentation that is valuable.

8. Final considerations

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 that foster knowledge sharing across agile software development teams. This model fosters awareness that inter-team knowledge sharing effectiveness is not only reached with a list of specific practices, but with a sustainable context in which the whole process is implemented.

Agile software development is now passing to 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. We provide some propositions (*P1* to *P14*) as implications for research that may be investigated in further studies to confirm or refute the issues influencing in

the inter-team knowledge sharing effectiveness. Beyond those propositions, other validations should be accomplished in future studies regarding (a) the intricate relationships between the proposed conditions and stimuli, (b) the proposed knowledge sharing purposes and (c) the practices applied specifically in agile contexts.

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

[1] J. T. Karlsen, L. Hagman, T. Pedersen, Intra-project transfer of knowledge in information systems development firms. Journal of Systems and Information Technology, vol. 13, no. 1, pp. 66-80, 2011.

[2] G. Ruhe, F. Bomarius, Learning Software Organization - Methodology and Applications. Lecture Notes in Computer Science Vol. 1756, Springer 2000.

[3] A. Boden, G. Avram, L. Bannon, and V. Wulf, Knowledge Management in Distributed Software Development Teams – Does Culture Matter? In: Proceedings of the Fourth IEEE International Conference on Global Software Engineering, pp. 18-27, 2009.

[4] T. Chau, F. Maurer, Knowledge Sharing in Agile Software Teams. In Logic versus Approximation, LNCS 3075, pp. 173-183. Springer-Verlag Berlin Heidelberg. 2004.

[5] I. Nonaka and H. Takeuchi, The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, Oxford, UK, 1995.

[6] L. Wah, Making knowledge stick. Management Review, vol. 88 no. 5, pp. 24-33, May 1999.

[7] R. K. Wagner and R. J. Sternberg, Tacit knowledge and intelligence in the everyday world. In Practical intelligence: Nature and origins of competence in everyday world. Cambridge University Press, 1986, pp. 51-83.

[8] R. Du, S. Ai, Y. Ren, Relationship between knowledge sharing and performance: A survey in Xi'an, China. Expert Systems with Applications 32: 38–46, 2007.

[9] K. Beck, and C. Andres, Extreme Programming Explained: Embrace Change. Addison-Wesley, 2nd ed. 2004.

[10] K. Schwaber, Agile Project Management with Scrum. Microsoft Press. Redmond, WA, 2004.

[11] T. Dingsøyr, T. Dybå and N. B. Moe, Agile Software Development - Current Research and Future Directions. Springer, 2010.

[12] P. Kettunen, Adopting key lessons from agile manufacturing to agile software product development—A comparative study. Technovation, 29(6-7):408–422, June 2009.

[13] V. Santos and A. Goldman, An Approach on Applying Organizational Learning in Agile Software Organizations. In: Agile Processes in SE and XP. Lecture Notes in Business Information Processing, vol. 77, no. 4, pp. 324-325, 2011.

[14] T. Kähkönen, Agile methods for large organizations - Building communities of practice. In Proceedings of the Agile Development Conference, pp. 2–11, IEEE Computer Society, Washington, DC, USA, 2004.

[15] F. O. Bjørnson, T. Dingsøyr, Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. Information and Software Technology, 50:1055–1068, 2008.

[16] A. Shalloway, G. Beaver, J. R. Trott, Lean-Agile Software Development: Achieving Enterprise Agility. Addison-Wesley Professional, 2009.

[17] J. N. Cummings, Work groups, structural diversity, and knowledge sharing in a global organization. Management Science, 50(3):352−364, 2004.
[18] M. T.Hansen, N. Nohria, and T. Tierny, What's Your Strategy for Managing Knowledge?, Harvard Business Review, vol. 77 , 106-116, 1999.
[19] K. E. Sveiby, People-Focused Knowledge Management: How Effective Decision Making Leads to Corporate Success. Butterworth-Heinemann, 2004.
[20] S. Sirkemaa, Challenge of knowledge sharing: Integrating customer in product development. Proceedings of the 8th WSEAS International Conference on Applied Computer Science (ACS'08). ISSN: 1790-5109, 2008.
[21] S. Wang, R. A. Noe, Knowledge sharing: A review and directions for future research. Human Resource Management Review, 20:115–131, 2010.
[22] P. Brössler, Knowledge Management at a Software Engineering Company – An Experience Report. Proceedings of the International Workshop on Learning Software Organizations (LSO), pp. 77-86, Kaiserslautern, Germany, Springer, 1999.
[23] V. Basili, M. Lindvall, and P. Costa, Implementing the Experience Factory Concepts as a Set of Experience Bases, Proceedings of the 13th International Conference on Software Engineering & Knowledge Engineering (SEKE), pp. 102-109, 2001.
[24] A. Cockburn, Crystal Clear: A Human-Powered Methodology for Small Teams. Addison-Wesley Professional, 2004.
[25] M. Levy, O. Hazzan, Knowledge management in practice: the case of agile software development. In Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering. Vancouver, Canada, 2009, pp. 60–65.
[26] V. Santos, A. Goldman, A. C. Shinoda, A. Fischer, A view towards Organizational Learning: An empirical study on Scrum implementation. In the Proceedings of the 23rd International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 583-589, 2011a.
[27] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, Manifesto for agile software development. http://agilemanifesto.org/, 2001.
[28] G. Von Krogh, K. Ichijo, I. Nonaka, Enabling knowledge creation: How to unlock the mystery of tacit knowledge and release the power of innovation. Oxford University Press, Oxford, UK, 2000.
[29] M. R. J. Qureshi, M. Kashif, Seamless Long Term Learning in Agile Teams for Sustainable Leadership. In Proceedings of the 5th IEEE International Conference on Emerging Technologies, pp. 389-394. 2009.
[30] T. Chau, Inter-Team Learning for Agile Software Processes. Master thesis. University of Calgary, Department of Computer Science, 2005.
[31] P. J. F. Treccani and C. R. B. de Souza, Collaborative Refactoring: Results of an Empirical Study Using Grounded Theory. CRIWG 2011, LNCS 6969, pp. 73–80, 2011.
[32] T. F. Fægri, T. Dybå, T. Dingsøyr, Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. Information and Software Technology, 52:1118–1132, 2010.
[33] K. Schwaber and M. Beedle, Agile Software Development with SCRUM. Prentice-Hall, 2002.
[34] J. Srinivasan, K. Lundqvist: Using Agile Methods in Software Product Development: A Case Study. In: Sixth International Conference on Information Technology: New Generations. 2009.

[35] Lindvall, Mikael and Basili, Victor R. and Boehm, Barry W. and Costa, Patricia and Dangle, Kathleen and Shull, Forrest and Tesoriero, Roseanne and Williams, Laurie A. and Zelkowitz, Marvin V. Empirical Findings in Agile Methods, Proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods - XP/Agile Universe 2002.

[36] E. Wenger, R. McDermott, and W. Snyder, Cultivating Communities of Practice: A Guide to Managing Knowledge. Boston: Harvard Business School Press, 2002.

[37] J. S. Brown and P. Duguid, Organizational learning and communities-of-practice: Toward a unified view of working, learning and innovation. Organization Science, 2(1), February 1991.

[38] A. Mestad, T. Dingsøyr, T. Dyba, 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, 2007.

[39] T. Kähkönen, Agile Methods for Large Organizations – Building Communities of Practice. Proceedings of the Agile Development Conference (ADC'04), 2004.

[40] K. Conboy and B. Fitzgerald,Method and Developer Characteristics for Effective Agile Method Tailoring: A Study of XP Expert Opinion. ACM Transactions on Software Engineering and Methodology, vol. 20, no. 1, article 2, 2010.

[41] V. A. Santos, A. Goldman, and C. D. Santos. Diagnosing Improvement Actions and Refining Research Design for the XP Laboratory Course. Submitted to CLEI Electronic Journal, Special Issue: ESELAW 2010, ISSN 0717- 5000, 2011b. (Waiting first assessment).

[42] K. C. Desouza, Facilitating tacit knowledge exchange. Communications of the ACM 46(6)85-88. June, 2003.

[43] Steven Fraser, Linda Rising, Scott Ambler, Alistair Cockburn, Jutta Eckstein, David Hussman, Randy Miller, Mark Striebeck, Dave Thomas. A Fishbowl with Piranhas Coalescence, Convergence or Divergence: The Future of Agile Software Development Practices Some Assembly Required!. Proceedings of the Conference on Object Oriented Programming Systems Languages and Applications OOPSLA, pp. 937-939, 2006.

[44] A. Strauss, Qualitative analysis for social scientists. New York, Cambridge University Press, 1987.

[45] B. G. Glaser, andA. L. Strauss, The Discovery of Grounded Theory: Strategies for Qualitative Research. New York, Aldine Publishing Company, 1967.

[46] M. Aniche and G. Silveira. Increasing Learning in an Agile Environment: Lessons Learned in an Agile Team. Proceedings of the Agile Conference, pp. 289 - 295. Salt Lake City, UT. August 2011.

[47] K. Beck, Test-Driven Development by Example. Addison Wesley. 2002.

[48] H. Corbucci, A. Goldman, E. Katayama, F. Kon, C. O. Melo, and V. Santos, "Genesis and Evolution of the Agile Movement in Brazil - A Perspective from the Academia and the Industry". Accepted for publication in*: SBES is 25,* Proc. of 25th Brazilian Symposium on Software Engineering, São Paulo, Brazil, September 2011.

[49] R. K. Yin, Case Study Research: Design and Methods Applied Social Research Methods. (Sage Publications, Inc., 4th edition, 2008).

[50] M. Dierkes, A. B. Antal, J. Child, and I. Nonaka, Handbook of Organizational Learning and Knowledge. Oxford Press, 2003.

[51] J. P. Kotter, The Heart of Change: Real-Life Stories of How People Change Their Organizations. Harvard Business Press, 2002.

[52] M. Easterby-Smith, M. A. Lyles. Handbook of organizational learning and knowledge management. Wiley, 2nd edition, 2011.

[53] P. F. Drucker, People and Performance: The Best of Peter Drucker on Management. Butterworth - Heinemann, UK, 1995.

[54] H. Mintzberg and A. McHugh. Strategy Formation in an Adhocracy. Administrative Science Quarterly, Vol. 30, No. 2., pp. 160-197, June 1985.
[55] C. M. Fiol, M. A. Lyles, Organizational Learning. The Academy of Management Review, 10 (4) 803–814, October 1985.
[56] I. Nonaka and G. von Krogh, Tacit knowledge and knowledge conversion: controversy and advancement in organizational knowledge creation theory, Organization Science, Vol. 20 No. 3, pp. 635-52, 2009.
[57] I. Nonaka and R. Toyama, Strategic management as distributed practical wisdom (phronesis), Industrial and Corporate Change, Vol. 16, pp. 371-94, 2007.
[58] I. Nonaka, R. Toyama, and N. Konno, 'SECI, *ba* and leadership: a unified model of dynamic knowledge creation, Long Range Planning, Vol. 33, pp. 5-34, 2000.
[59] I. Nonaka and N. Konno, The concept of *ba*: building a foundation for knowledge creation, California Management Review, Vol. 40, pp. 40-54, 1998.
[60] C. W. Choo and R. C. D. de Alvarenga Neto, Beyond the *ba*: managing enabling contexts in knowledge organizations, Journal of Knowledge Management, Vol. 14,Issue 4, pp.592 – 610, 2010.
[61] I. Nonaka, Creating Organizational Order out of Chaos: Self-Renewal in Japanese Firms, California Management Review, 30, 57-73, 1988.
[62] T. H. Davenport, L. Prusak, Working Knowledge: How Organizations Manage What They Know. Harvard Business School Press, Cambridge, MA, 1998.
[63] W. Lin, The effect of knowledge sharing model. Expert Systems with Applications.Vol. 34, 1508–1521, 2008.
[64] F. K. Y. Chang and J. Y. L. Thong, Acceptance of agile methodologies: A critical review and conceptual framework.In Decision Support Systems, vol. 46, no. 4, Mar. 2009, pp. 803-814.
[65] L. A. Joia and B. Lemos. Relevant factors for tacit knowledge transfer within organisations. Journal of Knowledge Management, vol. 14,no. 3, pp. 410-427, ISSN 1367-327, 2010.
[66] K. Chiri, J. Klobas, Knowledge Sharing and Organisational Enabling Conditions. In the Proceedings of the 11th European Conference on Knowledge Management (ECKM), 2010.
[67] M. J. Donate and F. Guadamillas, "Organizational Factors to Support Knowledge Management and Innovation", Journal of Knowledge Management, Vol. 15, Issue 6, 2011.
[68] H. Takeuchi and I. Nonaka, Hitotsubashi on Knowledge Management. Wiley, 2004.
[69] J. A. Highsmith, Adaptive Software Development: A Collaborative Approach to Managing Complex Systems. Dorset House, 1999.
[70] J. Highsmith, Adaptive Leadership – Accelerating enterprise agility. 2011.
[71] A. Cockburn, Agile Software Development: The Cooperative Game. Addison-Wesley Professional, 2nd Edition, 2006.
[72] N. Dixon, Common Knowledge: how companies thrive by sharing what they know. Harvard Business School Press, USA, 2000.
[73] J. Corbin and A. C. Strauss, Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Sage Publications, 3rd edition, 2007.
[74] D. T. Sato, H. Corbucci, and M. V. Bravo, "Coding dojo: An environment for learning and sharing agile practices". In the Proceedings of the Agile Conference, pp. 459–464, 2008.
[75] M. A. C. Silva, H. Roriz Filho, and H. F. N. Silva, Analysis of *Ba* during Scrum process. Proc. of 17th Symposium of Production Engineering (SIMPEP), pp. 1-15.Proj. Mangement and Production Engineering. Bauru, São Paulo, Brazil. November 2010 (In Portuguese).
[76] J. Liker, The Toyota Way. McGraw-Hill, USA, 2004.
[77] S. Gregor, The Nature of Theory in Information Systems. MIS Quarterly, vol.30, pp.611–642, 2006.

[78] R. P. Maranzato, M. Neubert, and P. Herculano, Moving back to scrum and scaling to scrum of scrums in less than one year. In Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion (SPLASH '11). ACM, New York, NY, USA, pp. 125-130, 2011.

[79] D. H. Pink, Drive: The Surprising Truth About What Motivates Us. Riverhead Trade, Reprint edition, 2011.

[80] C. Haythornthwaite, Social network analysis: An approach and technique for the study of information exchange. Library and Information Science Research, vol.18, pp. 323-342, 1996.

[81] A.M. Pettigrew, On studying organizational cultures. Administrative Science Quarterly, 570-581, 1979.