FEDERAL UNIVERSITY OF TECHNOLOGY - PARANÁ
CORNÉLIO PROCÓPIO
GRADUATE PROGRAM IN INFORMATICS

GLAUCO ANTONIO RUIZ

SYNTHESIZING RESEARCHES ON KNOWLEDGE MANAGEMENT AND AGILE SOFTWARE DEVELOPMENT USING THE META-ETHNOGRAPHY METHOD

DISSERTATION – MASTER

CORNÉLIO PROCÓPIO

2019

GLAUCO ANTONIO RUIZ

SYNTHESIZING RESEARCHES ON KNOWLEDGE MANAGEMENT AND AGILE SOFTWARE DEVELOPMENT USING THE META-ETHNOGRAPHY METHOD

Dissertation presented to the Postgraduate Program in Informatics of the Federal University of Technology - Paraná (UTFPR) as a partial requirement to obtain the title of "Master in Informatics".

Advisor: Prof.ª Dr.ª Érica Ferreira de Souza

CORNÉLIO PROCÓPIO

2019

Dados Internacionais de Catalogação na Publicação

R934 Ruiz, Glauco Antonio

Synthesizing researches on knowledge management and agile software development using the meta ethnograohy method / Glauco Antonio Ruiz. - 2019.
54 f. : il. ; 31 cm.

Orientador: Érica Ferreira de Souza.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Informática, Cornélio Procópio, 2019.
Bibliografia: p. 42-44.

1. Gestão do conhecimento. 2. Desenvolvimento de software. 3. Meta-etnografia. 4. Métodos ágeis. 5. Informática – Dissertações. I. Souza, Érica Ferreira, orient. II. Universidade Tecnológica Federal do Paraná. Programa de Pós-Graduação em Informática. III. Título.

CDD (22. ed.) 004

Biblioteca da UTFPR - Câmpus Cornélio Procópio
Bibliotecário/Documentalista responsável:
Simone Fidêncio de Oliveira Guerra – CRB-9/1276

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Programa de Pós-Graduação em Informática

Título da Dissertação Nº 68:

"SYNTHESIZING RESEARCHES ON KNOWLEDGE MANAGEMENT AND AGILE SOFTWARE DEVELOPMENT USING THE META-ETHNOGRAPHY METHOD".

por

Glauco Antonio Ruiz

Orientadora: **Profa. Dra. Érica Ferreira de Souza**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM INFORMÁTICA – Área de Concentração: Computação Aplicada, pelo Programa de Pós-Graduação em Informática – PPGI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 15h00 do dia 02 de dezembro de 2019. O trabalho foi ____________ pela Banca Examinadora, composta pelos professores:

Profa. Dra. Katia Romero Felizardo Scannavino
(Presidente – UTFPR-CP)

Prof. Dr. Willian Massami Watanabe
(UTFPR-CP)
Participação à distância via ______________

Prof. Dr. João Coelho Neto
(UENP)

Visto da coordenação:

Danilo Sipoli Sanches
Coordenador do Programa de Pós-Graduação em Informática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

Dedico essa dissertação primeiramente a Deus por me dar saúde, determinação e discernimento para mais uma conquista antes inimaginável.

A minha amada esposa Patricia por sempre ter a paciência e compreensão necessários para o cumprimento dessa e de outras etapas de nossas vidas, sempre incentivando, orando e me protegendo de todas as formas possíveis, valorizando cada passo conquistado por nós. Sem você nada seria e nunca será possível. Essa é mais uma conquista nossa. Te amo!

Aos meus amados filhos, noras e netos que sempre estiveram comigo nessa caminhada, obrigado pelo apoio e compreensão nas minhas faltas. Renan, Leticia, Vitor, Glauco Jr, noras e noro, Giovanna e Vinícius, amo vocês demais.

A toda minha família, em especial minha irmã Claudete, que foi a razão de tudo isso começar lá atrás nos primórdios e por ter me incentivado a ser a pessoa diferente que hoje sou. Você é um guia, um alicerce de toda a família.

AGRADECIMENTOS

Meus sinceros agradecimentos a minha orientadora Profª Drª Érica Ferreira de Souza por ter me aceito, orientado, me cobrado e, principalmente, me ajudado. Sem a sua dedicação eu jamais teria conseguido, você é um exemplo de docência. Obrigado de coração.

Aos meus amigos do grupo de estudos PNC: Marco, Eduardo, Alex, André e Patrick pelas viagens, estudos, trabalhos e parcerias que tivemos durante toda a jornada.

A minha parceira prodígio de artigos, sempre muita dedicada, prestativa e inteligente. Voa Bianca Napoleão, você merece todo o sucesso que está alcançando.

A todos que me incentivaram ou duvidaram que eu pudesse atingir esse objetivo, principalmente a quem tentou obstruir essa caminhada. Esse foi meu combustível.

Esta dissertação de mestrado fez parte do projeto de pesquisa CNPq Universal (número 432247/2018-1). Todos os envolvidos agradecem o apoio financeiro recebido.

“Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes”.

Martin Luther King

ABSTRACT

RUIZ, GLAUCO. SYNTHESIZING RESEARCHES ON KNOWLEDGE MANAGEMENT AND AGILE SOFTWARE DEVELOPMENT USING THE META-ETHNOGRAPHY METHOD. 54 f. Dissertation – Master – Graduate Program in Informatics, Federal University of Technology - Paraná. Cornélio Procópio, 2019.

Context: Software development processes are considered as knowledge intensive and therefore Knowledge Management (KM) can be applied to efficiently manage the knowledge generated. Agile practices can benefit the software organizations in terms of KM. Some studies have already presented evidence about this relationship. However, the intersection of these two areas still require further clarification. **Objective:** This study aims to synthesize research on KM and Agile Software Development (ASD) using the meta-ethnography method considering Scrum and XP frameworks as scope. **Method:** In order to achieve the proposed goal, first, the phases of the meta-ethnography analysis method were applied to a set of articles selected from a tertiary review of KM and ASD, as well as classic guides and area references. Finally, the relationships identified between the investigated areas were analyzed through interviews with experts in agile development methodology. **Results:** KM activities during ASD can improve team learning and collaborate with the evolution of organizational knowledge. The most common activity investigated between KM and ASD is knowledge sharing. Applying meta-ethnography and confirming the synthesization of interviews with ASD specialists, it was possible to identify relationships and associations between agile values, Scrum elements and main XP practices within the integrated KM cycle. **Conclusion:** The clarification of how KM is present in each agile value, practices and artifacts allows a reflection on how much knowledge was created, shared and applied during ASD. In addition, such results presented in this study enable organizations to know each other better and to explore more each KM activity, thus contributing to delivering more value to the customer.

Keywords: Meta-ethnography, Knowledge Management, Agile Software Development, Scrum, XP

RESUMO

RUIZ, GLAUCO. SYNTHESIZING RESEARCHES ON KNOWLEDGE MANAGEMENT AND AGILE SOFTWARE DEVELOPMENT USING THE META-ETHNOGRAPHY METHOD. 54 f. Dissertation – Master – Graduate Program in Informatics, Federal University of Technology - Paraná. Cornélio Procópio, 2019.

Contexto: Os processos de desenvolvimento de software são considerados intensivos em conhecimento e, portanto, a Gestão do Conhecimento (GC) pode ser aplicada para gerenciar com eficiência o conhecimento gerado. As práticas ágeis podem beneficiar as organizações de software em termos de GC. Alguns estudos já apresentaram evidências sobre essa relação. No entanto, a interseção dessas duas áreas ainda requer esclarecimentos adicionais. **Objetivo:** Este estudo tem como objetivo sintetizar pesquisas sobre GC e Desenvolvimento Ágil de Software (DSA), utilizando o método da meta-etnografia, considerando os *frameworks* Scrum e XP como escopo. **Método:** Para atingir o objetivo proposto, primeiro, as fases do método de análise meta-etnográfica foram aplicadas a um conjunto de artigos selecionados a partir de uma revisão terciária de GC e DSA, além das referências clássicas da área. Por fim, as relações identificadas entre as áreas investigadas foram analisadas por meio de entrevistas com especialistas em DSA. **Resultados:** As atividades de GC durante o DSA podem melhorar o aprendizado em equipe e colaborar com a evolução do conhecimento organizacional. A atividade mais comum investigada entre GC e DSA é o compartilhamento de conhecimento. Aplicar a meta-etnografia e confirmar as sintetizações com entrevistas feitas junto a especialistas de DSA, foi possível identificar relacionamentos e associações entre valores ágeis, elementos do Scrum e principais práticas do XP junto ao ciclo integrado da GC. **Conclusão:** O esclarecimento de como a GC está presente em cada valor, práticas e artefatos ágeis permite refletir sobre quanto conhecimento foi criado, compartilhado e aplicado durante o DSA. Além disso, os resultados apresentados neste estudo permitem que as organizações se conheçam melhor e explorem cada atividade de GC, contribuindo assim para agregar mais valor ao cliente.

Palavras-chave: Meta-etnografia, Gestão do Conhecimento, Desenvolvimento de Software Ágil, *Scrum*, XP

FIGURES

TABLES

ACRONYMS

KM	Knowledge Management
ASD	Agile Software Development
SE	Software Engineering
RQ	Research Question
XP	Extreme Programming
PO	Product Owner
SLRs	Systematic Literature Reviews
IC	Inclusion Criteria
EC	Exclusion Criteria

CONTENTS

1 INTRODUCTION

A challenge of Software Engineering (SE) is the systematic strategies creation in order to integrate the knowledge involved in a project. Software organizations seek for solutions that emphasize different knowledge types when planning initiatives to manage them (BJØRNSON; DINGSØYR, 2008). A systematic routine for capturing knowledge is important in an organization, making knowledge transparent to everyone involved to be benefited from accumulated knowledge (LEVY; HAZZAN, 2009). However, while the importance of working with the knowledge has been widely recognized in many areas, managing that knowledge is still a challenge for software development organizations. Knowledge Management (KM) principles are mechanisms that can at least partially solve such challenges (DAVENPORT; PRUSAK, 2000).

KM aims to promote knowledge creating, sharing, storage and make organizational knowledge accessible and reusable within the organization (O'LEARY; STUDER, 2001). According to O'Leary (1998), KM formally manages knowledge resources to facilitate access and reuse. Nonaka e Takeuchi (1995) use the tacit-explicit distinction to differentiate inarticulate and articulate knowledge. Tacit knowledge is knowledge based on experience that cannot be documented and usually remains only in people's minds. Tacit knowledge encompasses knowledge associated with the senses, skills, experiences or intuition. Explicit knowledge, in turn, represents knowledge that can be documented as long as it is objective and rational. Explicit knowledge can be easily used and shared.

Traditional software development involves using multiple documents to capture and represent knowledge related to the various stages of software development lifecycles (WENDORFF; APSHVALKA, 1998). In this traditional scenario, explicit knowledge is decisive. Unlike traditional software methods, Agile Software Development (ASD) is considered a lightweight software development. Agile frameworks emphasize collaboration among team members in applying and sharing knowledge. Agile frameworks prioritize tacit knowledge, encouraging individual, team, and customer communications and interactions (ANDRIYANI et al., 2017). Therefore, agile practices and KM present common activities that can benefit software organizations to promote knowledge sharing, team communication, knowledge reuse, and the collaborative process. However, according to Cabral et al. (2014), there is still a gap about what emerges from the intersection of these two areas requiring further clarification. For this reason, KM in ASD has been treated as a broad research topic, resulting in various relationships between its schools and concepts.

1.1 OBJECTIVE

Based on the above context, we intend to understand how ASD is inherent to KM activities. We decided to explore the following Research Question (RQ) to guide us to evidence the relations in both these areas:

How do agile values and practices relate to knowledge management activities?

1.2 RESEARCH METHOD

In this project, a study to synthesize events and artifacts of the Scrum and XP frameworks with KM was conducted. In this study, the seven phases of meta-ethnographic method were applied in eigth articles selected from a review of secondary studies in KM and ASD. Through the seven applied phases it was possible to identify the main concepts and relationships between the main KM activities (create, share and apply the knowledge) with the Scrum framework artifacts and activities as well as with XP practices. In addition, the concepts and relationships identified between the areas investigated were validated based on interviews with four ASD professionals.

The main activities conducted in this dissertation were:

1. A tertiary study to understand the state of the art on how KM activities have been applied in the ASD and assemble an initial set of candidate studies considered as input to meta-ethnography;

2. Conduction of the synthesis process proposed by the meta-ethnography method considering the two main methods of ASD: Scrum and Extreme Programming (XP); and

3. Conducting interviews with professionals who present a practical view on ASD in order to validate the identified relationships between KM and ASD.

1.3 TEXT ORGANIZATION

The remainder of this document is structured as follows. Section 2 reviews the literature of KM, ASD, Meta-ethnography Method and Related Works. Section 3 presented the application of Meta-ethnography Method. Finally, conclusions remarks, limitations and future directions are described in Section 4.

2 BACKGROUND

In this chapter, the main concepts of this dissertation are discussed.

2.1 KNOWLEDGE MANAGEMENT

KM is the process through which organizations generate value from their intellectual assets (tacit and explicit knowledge) (BUKOWITZ; WILLIAMS, 2000). Most often, generating value from such assets involves sharing them among employees, departments and even with other companies in an effort to devise best practices.

KM also can be defined as a set of organizational activities that must be performed in a systematic manner. Effective KM requires an organization to execute activities such as to identify, generate, acquire, diffuse, and capture knowledge. The way how activities are organized are called KM cycles (also known as KM models) (DALKIR, 2005).

KM cycles can be used to guide how to conduct KM activities. KM cycles have an objective to help identifying and locating knowledge and knowledge sources within the organization. According to Dalkir (2005), the four most known KM cycles are presented by Wiig (1997), Meyer e Zack (1996), (LEVY; HAZZAN, 1999) and Bukowitz e Williams (2000). These KM cycles identify and locate knowledge and knowledge sources within the organization from several activities. However, Dalkir (2005) also mentions that a similar lack of consensus exists with respect to the terms used to describe the major activities in the KM cycle. So, on the basis of main KM cycles, Dalkir (2005) one can distill an integrated KM cycle represented by interaction of three activities:

- **Knowledge capture and/or creation:** Knowledge capture refers to the identification of existing internal and/or external knowledge from the environment. Knowledge creation is the development of new knowledge that did not have a previous existence within the organization. In this activity the tacit knowledge is captured or elicited, and explicit knowledge is organized or coded.

- **Knowledge sharing and dissemination:** Once knowledge has been captured and coded, it needs to be shared and disseminated throughout the organization. There are several practices that can be used to share knowledge within the organization, such as team meetings, written instructions, ad hoc, verbally, intranet or video.

- **Knowledge acquisition and application:** Knowledge that has been captured, coded, shared, and otherwise made available is put to use. KM can succeed only if the knowledge is used. However, it now becomes imperative to understand which knowledge is of use to which set of people and how best to make it available. The use of KM systems, such as expertise location systems or content management systems can be designed to optimize knowledge application on an organization-wide basis.

The transition between the three activities is represented in Figure 1. In integrated KM cycle, the transition from knowledge capture/creation to knowledge sharing and dissemination, knowledge content is assessed. Knowledge is then contextualized in order to be understood ("acquisition") and used ("application"). This process then feeds back into the first activity in order to update the knowledge content (DALKIR, 2005).

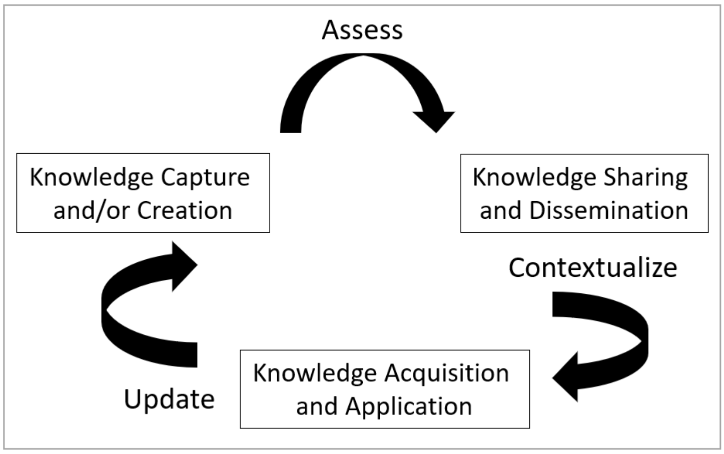
Assess
Knowledge Capture and/or Creation
Knowledge Sharing and Dissemination
Contextualize
Update
Knowledge Acquisition and Application

Figure 1: An integrated KM Cycle

Source: Adapted from (DALKIR, 2005)

In this dissertation, the integrated KM cycle presented in Dalkir (2005), was used to represent the main KM activities serving as input for the synthesis method presented in Section 2.4.

2.2 KNOWLEDGE MANAGEMENT AND SOFTWARE ENGINEERING

One of the main characteristics of SE is the high volume of information that is generated and manipulated in the organization. Those involved in the project face problems such as: difficulty in systematizing the information generated throughout the software processes; difficulty in reusing knowledge generated from one project in another; loss of organization intellectual capital; and the non-representation of knowledge (SOUZA et al., 2015). From these problems, integrating KM in SE has brought much discussion about how to manage knowledge in the organization. According to Bjørnson e Dingsøyr (2008), research related which KM and SE focuses, specially, on ways and means to share knowledge.

Knowledge in a SE organization should be properly captured, stored and reused when needed. KM principles and techniques application aims at facilitating knowledge flow and utilization across SE process. In this environment knowledge needs to be updated all the time since software development environment technologies used are often changing (VASANTHAPRIYAN et al., 2015).

In SE, there is a lot of discussion going on about how to manage the knowledge, or how to promote “Learning Software Organizations”. When organizations talk about KM, one of the main practices is to develop an “Experience Factory”. In a software development environment, the experience of each activity conducted can be collected, packed and stored in a knowledge base to be easily reused, documented and, therefore, accessed by several members of the organization. The explicit knowledge can be described by drawings and writings, consequently, it can be easily used and shared (BJØRNSON; DINGSØYR, 2008).

Vasanthapriyan et al. (2015) describe that an effective management of software development process became a need for the software development organizations to survive on the competitive scenario. In order to get strengths on the development process business, the organizations need to execute the software development efficiently, introducing KM principles that support the software process. Therefore, introducing KM in SE concepts is essential. Beyond that, KM helps the organization to improve the decision-making process, improving innovation and organization performance and also helping the organization to sustain the competitiveness.

KM principles are present in software development organizations, from the organizations, which use traditional practices as well as the ones that use agile practices. Regarding those with agile practices, lots of KM activities can be considered resembling, for example, the communication and knowledge sharing (CABRAL et al., 2014). The study about KM activities in agile practices might help development teams to improve learning and to collaborate with the organization knowledge evolution, leading to high value deliveries and, consequently, raising the customer satisfaction.

2.3 AGILE SOFTWARE DEVELOPMENT

Traditional software development process can limit developers, since it sometimes turns into a complex and expensive process. Besides, it is extremely focused on documentation. This fact highlights the emergence of agile methodologies (SCHÖN et al., 2015). Projects using agile methodologies assume that change is common in software projects (and software-heavy projects) and thus value ongoing planning, emphasizing human aspects and adaptability to rapid changes in the project.

In 2001, leaders of different streams joined hands and created the Manifesto for ASD (BECK et al., 2001). Agile Manifesto includes values and principles that help to optimize the software development process and also have a strong influence on present practices of team collaboration within ASD (SCHÖN et al., 2015). The Agile Manifesto provides four core values:

(I) Individuals and interactions over processes and tools;

(II) Working software over comprehensive documentation;

(III) Customer collaboration over contract negotiation; and

(IV) Responding to change over following a plan.

There are many frameworks in ASD, and among the best-known and most commonly used are Scrum and XP (CABRAL et al., 2014) frameworks. In order to represent the relation of agile practices with KM, in this dissertation we use Scrum and XP frameworks as input to the synthesis method. We chose Scrum and XP as an input for this work since they are currently considered the most commonly used frameworks for software development (COBB, 2015). Following, these two frameworks are briefly presented.

2.3.1 SCRUM

Scrum Guide (SCHWABER; SUTHERLAND, 2017) official document defines Scrum as a framework for developing, delivering, and sustaining complex products. According to Sutherland e Schwaber (2012), the guide focuses on two core elements: Scrum events (more precisely in Sprint event, which is a container for all other events) and Scrum artifacts. These elements are briefly presented below

Scrum Events. Prescribed events are used in Scrum to create regularity and to minimize the need for meetings not defined. The events are specifically designed to enable critical transparency and inspection. Each event in Scrum is a formal opportunity to inspect and adapt something. One of the main events in Scrum is Sprints. Sprint is a process that lasts on average one month to deliver an incremented version. Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective (SUTHERLAND; SCHWABER, 2012; SCHWABER; SUTHERLAND, 2017).

The work to be done at Sprint is defined in the Sprint planning meeting. This planning is done by the entire Scrum team collaboratively. Daily Scrum, Scrum's daily meetings, are meetings that last fifteen minutes at most for the development teams to synchronize activities and build up a plan for the next 24 hours. According to Pressman (2006), during the meeting, each member of the development team clarifies the following questions: (i) What have you done since the last team meeting?; (ii) What obstacles are you facing?; and (iii) What do you plan to do until the next team meeting?

The Sprint Review Meeting is performed after the Sprint to verify the increase and to adapt to the backlog of the product if necessary. The Sprint Retrospective is an opportunity for the scrum team to inspect and create a plan for improvements that should be valid for the next Sprint (SUTHERLAND; SCHWABER, 2012).

Scrum Artifacts. Scrum artifacts differ from common artifacts because they are designed to provide transparency and enable inspection and adaptation ensuring that everyone has the same understanding about a particular artifact. Product Backlog is an orderly list of everything that can be necessary for the product and it is the only source for the requirements of any needed changes to be made in the product. The Product Backlog lists all features, every functionality, functions, requirements, improvements and repairs that represent changes to be made for the next version of the product. Its items have attributes descriptions, order and estimation of effort. Generally, it is ordered by the value it aggregates to the business, in terms of risk, priority and need. The very first items of the Product Backlog determine more immediate development activities (SUTHERLAND; SCHWABER, 2012; SCHWABER; SUTHERLAND, 2017).

Sprint Backlog is a set of the Product Backlog items selected for a Sprint and for a plan to obtain a product increase and to achieve the Sprint goals. Sprint Backlog defines the work to be done by the development team in order to make the Product Backlog items into 'Ready' increases (SUTHERLAND; SCHWABER, 2012). At the end of a Sprint, the new increment has to be 'Ready', which means that it is in a usable condition and meets the definition 'Ready' of the Scrum Team.

The Scrum Team is composed by the Product Owner (PO), the Development Team and the Scrum Master. Those teams are self-organized, and choose the best way to finish the work. The teams are also cross-functional and multi-functional and all of them have the necessary skills to carry out the work without dependence on others who are not part of the team.

The framework scrum is illustrated in Figure 2.

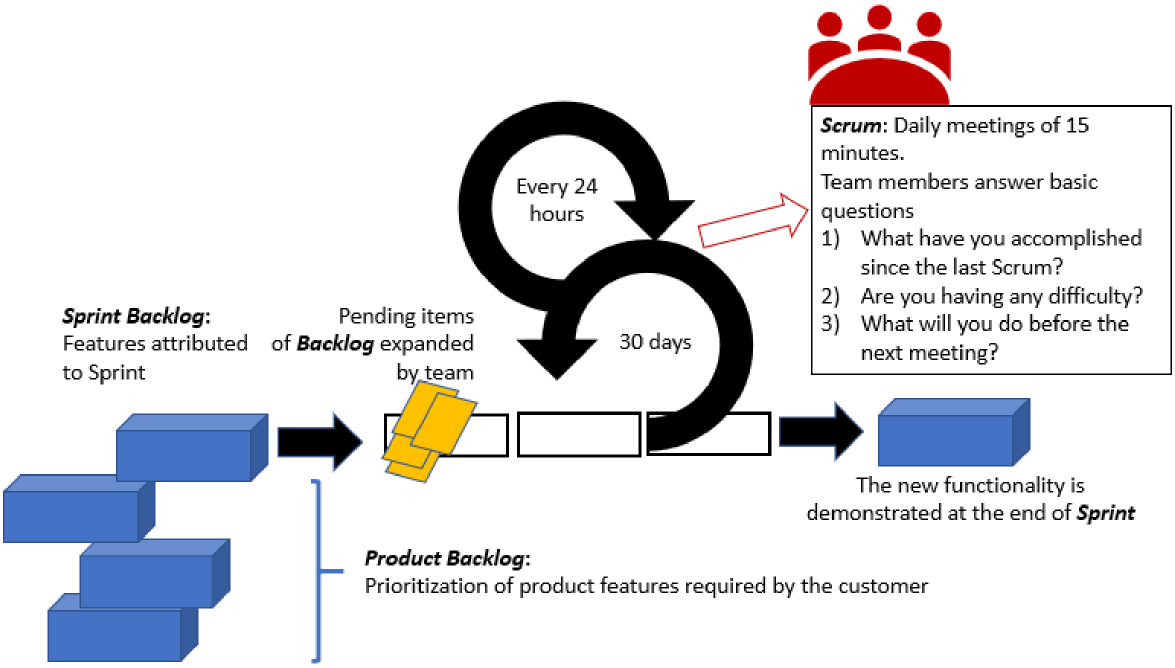
Every 24 hours
30 days
Sprint Backlog: Features attributed to Sprint
Pending items of Backlog expanded by team
Scrum: Daily meetings of 15 minutes.
Team members answer basic questions
1) What have you accomplished since the last Scrum?
2) Are you having any difficulty?
3) What will you do before the next meeting?
The new functionality is demonstrated at the end of Sprint
Product Backlog: Prioritization of product features required by the customer

Figure 2: Framework Scrum

Source: Adapted from (PRESSMAN, 2006)

2.3.2 EXTREME PROGRAMMING (XP)

XP is a style of software development focusing on application of programming techniques, clear communication, and teamwork that allows the software development based on the values of communication, feedback, simplicity, courage, and respect (BECK; ANDRES, 2004). XP framework uses the object-oriented approach. It contemplates a set of rules and practices in four activities: planning, design, codification and testing (PRESSMAN, 2006).

The planning activity consists in creating a set of stories from users who describe a set of characteristics and functionalities required for the software to be developed. The customers and developers work together to decide how to group up stories for the next increase. After the first increase is finished, the XP team calculates the project speed, which means the amount of stories implemented in the first version. With this, it is possible to estimate the delivery time and schedule the following versions. As the project moves, the customer can add new stories (PRESSMAN, 2006; BECK; ANDRES, 2004).

A simple project is desirable over a complex project. Thus, the XP follows the KIS principle (keep it simple) strictly. The project provides implementation guidelines to a story as it is written, nothing more, nothing less. XP encourages the use of CRC cards (Class – Responsibility – Collaborator) which identify and organize the stories that are putting the cards on a prominent wall (BECK; ANDRES, 2004). XP also encourages refactoring, which consists in an enhancement of the code after it is written (PRESSMAN, 2006).

A key-concept during the codification activity at XP is the programming in pairs. Pair programming is a dialog between two people simultaneously programming the story code (analyzing and designing and testing) (BECK; ANDRES, 2004).

It is important to emphasize the collective ownership of the source code in XP. The source code of a program has no exclusive owner and there is no need for a formal permission to make any modifications. Even though control tools are necessary, so that work is not lost, this XP practice allows the entire team to know every part of the system (PRESSMAN, 2006).

Another point that deserves to be highlighted is the continuous integration. Each integration or functionality made for the software must be integrated immediately to the current version of the system, thus, minimizing conflict problems among older versions and enabling the awareness of the real development status (BECK; ANDRES, 2004).

XP framework recommends creating a series of unitary tests, before the codification, which will exercise every single story that must be included in the actual version, because once those tests are created, the developer knows what is needed to be done to pass on the unitary test. These unitary tests are created in a way that they can be automated and easy to be executed, because they will be executed over and over again. Finally, the acceptation tests, or customer tests, are used by the customers based on their stories, focusing on the characteristics and functionalities of the global system that are visible and subject to change (PRESSMAN, 2006; BECK; ANDRES, 2004).

Figure 3 presents the XP workflow.

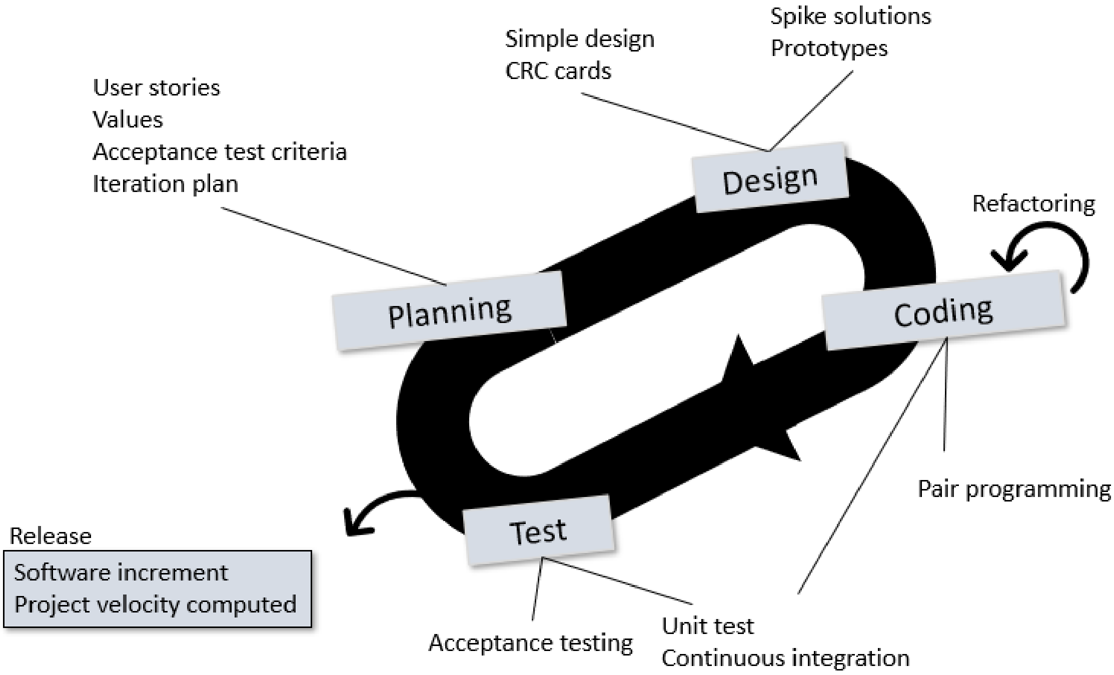

Figure 3: eXtreme Programming (XP) workflow

Source: Adapted from (PRESSMAN, 2006)

2.4 META-ETHNOGRAPHY METHOD

Noblit e Hare (1988) define meta-ethnography as an intensive synthesis method involving observation, interviewing and document review. It enables a systematic and detailed understanding of how studies are related. This relation is done through the comparison of findings within and across studies. Meta-ethnography is the method of qualitative synthesis most widely used in health and education research. According to Noblit e Hare (1988), a meta-ethnography could be identified as a study rather than a method since it helps a researcher to build an interpretative rather than an aggregate description of the evidence being synthesized.

In summary, researchers select, analyze and interpret qualitative studies though a process of translation, in which the studies are coded into metaphors, which provide an interpretation of the entire topic, in order to answer focused questions on a specific topic and gain new insights.

The meta-ethnography method is a qualitative analysis method composed of seven steps, briefly described below:

1. Getting started – the starting point involves identifying a topic that qualitative research

might inform. It is relevant to find a topic that could be of interest to both researchers and also for practitioners. Researchers can define a research question to represent the topic and guide the meta-ethnography application.

2. Deciding what studies are relevant to the topic of interest – the goal of the second stage is to find and select studies that are relevant to the topic of interest. It involves activities such as: searching candidate studies; making decisions on inclusion/exclusion; and quality assessment.

3. Reading the studies – during this stage researchers carefully read the set of included studies becoming as familiar as possible with their general content and detail. Researchers can also incorporate data extraction.

4. Determining how the studies are related – initially, researchers can list the key metaphors of each study, be phrases, ideas and/or concepts. Next, the researchers can create a list of these metaphors, juxtaposing them and determining how they are related. Grids or tables can also be used to display metaphors across all studies.

5. Translating the studies into one another – in this stage the goal is to describe how to compare metaphors in one study with those in another. The original method does not define exactly how to do this. One suggestion is to compare the metaphors from study 1 with study 2, and the synthesis of these two studies with study 3, and so on (SILVA et al., 2013a).

6. Synthesizing the translations – when the previous stage results in many metaphors, these can be compared to see if there are common types or if some metaphors can encompass others. In addition, the findings of the synthesis can be represented as diagrams or figures.

7. Expressing the synthesis – the findings of the synthesis can be finally disseminated to interested parties.

Meta-ethnography is often confused with other qualitative methods (synthesis) because of similarity between them (FU et al., 2019). Examples of other qualitative methods are Thematic Synthesis and Grounded Theory. The difference is that Thematic Synthesis is an extension of thematic analysis, which is often used to identify, analyze, and report patterns within data in primary research, and further creates a model of higher-order themes with descriptive summaries (CRUZES; DYBå, 2011a). On the other hand, meta-ethnography provides a new interpretation that goes beyond all the primary or a picture of the whole phenomenon under study from studies of its parts. Grounded theory further generates a new theory which can explain the targeted phenomenon using the data from primary studies (HANNES; LOCKWOOD, 2011).

2.5 RELATED WORK

In Fu et al. (2019) and Cruzes e Dybå (2011b) a tertiary studies were conducted on meta-ethnography used in Systematic Literature Reviews (SLRs) in SE. Both studies present the several benefits of meta-ethnography, but also highlight a general lack of attention paid to research synthesis in SE. The authors reported that there were only a few SLRs that used meta-ethnography method.

Fu et al. (2019), for example, mention that SLRs that appear to have used meta-ethnography for data synthesis have used some featured techniques of meta-ethnography rather than the complete defined method consisting of seven steps. In addition, some reviews in which authors did not explicitly state the use of meta-ethnography or did not know they were using it. There are some relationships between SLRs and meta-ethnography. So, some authors use SLRs to identify the main evidence on a particular research topic and then they use meta-ethnography to identify the relationships and connections among the evidence in Data Extraction and Data Synthesis activities of the SLRs.

As our dissertation aims to synthesize what has been researched between two distinct areas through meta-ethnography considering KM, we present below the only work found so far that explicitly presents a synthesis of KM with other area using the seven steps of meta-ethnography.

Silva et al. (2013b) explore the use of meta-ethnography in the synthesis of empirical studies in KM through an example using studies on the relationships between personality and software team processes. Thus, the seven phases of meta-ethnography were applied in a set of articles selected from a systematic review previously developed to evaluate the adequacy of meta-ethnography in this domain with respect to the ease of use, utility and reliability of the results.

Common concepts were identified through reading and interpretation of the studies. Then, second order translations have been built and used to synthesize a relation model between software team's personality and processes (SILVA et al., 2013b). Meta-ethnography is adequate in the synthesis of empirical studies, even in the context of studies of mixed methods. Silva et al. (2013b) also argue that the researcher who is dedicated to the use of meta-ethnography should be aware that the set of studies to synthesize can greatly influence the consistency and the reliability of the resulting synthesis.

Similarly as Silva et al. (2013b), in our study used meta-ethnography to synthesize the relationships that exist between two distinct areas. However, the areas investigated are Agile and KM, which we believe have strong relationships in the day to day execution of an Agile team.

3 APPLICATION OF META-ETHNOGRAPHY METHOD

In this chapter, the application of the meta-ethnography method was presented, considering the areas of ASD and KM. The meta-ethnography was constructed using the seven stages suggested by Noblit and Hare (NOBLIT; HARE, 1988). The results of applying these phases are presented in the following sections.

3.1 PHASE 1 - GETTING STARTED

In this study, we intend to understand how agile values and practices are inherent to KM activities. We want to perform a synthesis about the intersection of these two areas that needed further clarification for researchers and practitioners interested in the study of ASD and KM. Regarding ASD, we focus on Scrum and XP frameworks. With respect to Scrum, we decided to refine and explore the events and artifacts. In the case of XP, our focus was the XP practices.

3.2 PHASE 2 - DECIDING WHAT IS RELEVANT TO THE INITIAL INTEREST

In order to decide which studies are relevant to the initial interest we conducted a tertiary study looking for secondary studies investigating the state of the art in KM and agile development. Tertiary studies are considered as a review that focuses only on secondary studies, i.e., it is a review about other secondary studies (KITCHENHAM; CHARTERS, 2007). This review involves three main phases (KITCHENHAM; CHARTERS, 2007; FELIZARDO et al., 2017): (i) **Planning**: refers to identifying a need for conducting the review, and aims at establishing a review protocol defining the research questions, inclusion and exclusion criteria, sources of studies, search string; (ii) **Conducting**: searches and selects the studies, in order to extract and synthesize data from them; and (iii) **Reporting**: final phase that aims at writing up the results and circulating them to potentially interested parties.

An early version of this tertiary review was published in (RUIZ et al., 2018). The first version was conducted until December 2017 and five studies were identified to compose the initial set to conduct the research on synthesizing the areas. Posteriorly, an update of this tertiary review was conducted until April 2019 and three new studies were identified, totaling eight studies for this analysis.

We used the search string shown in Table 1. The string was applied in three metadata fields: title, abstract and keywords. The search string had syntax adaptations according to particularities of each database source.

Table 1: Keywords of the Search String of the Tertiary Study in KM and agile development.

Areas	Keywords
Agile	"agile", "extreme programming", "xp method", "scrum", "crystal", "dsdm", "fdd", "feature driven development", "lean software development"
KM	"Knowledge Management", "knowledge sharing", "knowledge transfer", "knowledge extraction", "Knowledge discovery", "useful knowledge", "tacit knowledge", "explicit knowledge", "knowledge creation", "knowledge acquisition", "knowledge retention", "knowledge evaluation", "knowledge application", "organization knowledge", "knowledge engineering", "Knowledge representation")
Review	"Systematic Literature Review", "Systematic Review", "Systematic Mapping", "Mapping Study", "Mapping Studies", "Systematic Literature Mapping", "Literature Review", "literature analysis"
Search String: ("agile" OR "extreme programming" OR "xp method" OR "scrum" OR "crystal" OR "dsdm" OR "fdd" OR "feature driven development" OR "lean software development") AND ("Knowledge Management" OR "knowledge sharing" OR "knowledge transfer" OR "knowledge extraction" OR "Knowledge discovery" OR "useful knowledge" OR "tacit knowledge" OR "explicit knowledge" OR "knowledge creation" OR "knowledge acquisition" OR "knowledge retention" OR "knowledge evaluation" OR "knowledge application" OR "organization knowledge" OR "knowledge engineering" OR "Knowledge representation") AND ("Systematic Literature Review" OR "Systematic Review" OR "Systematic Mapping" OR "Mapping Study" OR "Mapping Studies" OR "Systematic Literature Mapping" OR "Literature Review" OR "Literature Analysis")	

The database sources used in this tertiary review were: Scopus, IEEE Xplore Digital Library, Science Direct and Engineering Index Compendex. These databases are most commonly used in Computer Science (DYBA; DINGSOYR, 2008; MAPLESDEN et al., 2015). The selection criteria are organized in one Inclusion Criteria (IC) and five Exclusion Criteria (EC).

The inclusion criteria is:

(IC1) The study discusses KM and ASD methodology.

The exclusion criteria are:

(EC1) The study is just published as an abstract;

(EC2) The study is not written in English;

(EC3) The study is an older version of other study already considered;

(EC4) The study is not a secondary study, such as primary study or editorials, summaries of keynotes, workshops, and tutorials; and

(EC5) The study is not available.

A total of 205 studies were identified during the search process. First, we eliminated duplicated studies (publications that appear in more than one source), achieving 186 publications. Out of these studies, we selected secondary studies by reading their titles and abstracts and applying the inclusion and exclusion criteria. As a result, a total of 25 studies were selected. Next, the selection criteria were applied considering the full text. A total of eight studies were returned in this stage. Over these eight studies considered relevant, we performed backward snowballing. Snowballing was applied in order to identify additional relevant studies through the reference lists of the eight studies. However, no relevant study was identified. Thus, our initial set of studies was composed by eight relevant studies from the four sources that we searched, presented in Table 2.

In Table 2 the article reference, year of publication, how many primary studies were included in each study and the title are shown. It is worth mentioning that although the initial set is eight secondary studies, this set increased considering the primary studies returned in each one. The total of 238 primary articles were returned, which were analyzed when necessary.

3.3 PHASE 3 - READING THE STUDIES

Firstly, an individual reading was done by each author to identify the relations between “Agile values and KM”, “Scrum and KM” and “XP and KM”. Subsequently, we conducted several meetings in groups to resolve in consensus the divergences. Following is a brief presentation of the initial set of studies.

Analyzing the eight selected papers over the years, the intention to map and begin to correlate KM initiatives in ASD is recent, starting basically in 2011, as Table 2 suggests.

Table 2: Initial set of selected studies

Reference	Year	Nº Primary Study	Title study
(NEVES et al., 2011)	2011	14	Knowledge creation and sharing in software development teams using Agile methodologies: key insights affecting their adoption
(CABRAL et al., 2014)	2014	25	Knowledge Management in Agile Software Projects: A Systematic Review
(PAREDES et al., 2014)	2014	50	Information Visualization for Agile Software Development Teams
(ANDRIYANI et al., 2017)	2017	48	Understanding Knowledge Management in Agile Software Development Practice
(BORREGO et al., 2017)	2017	42	Review of approaches to manage architectural knowledge in Agile Global Software Development
(INDUMINI; VASANTHAPRIYAN, 2018)	2018	12	Knowledge Management in Agile Software Development - A Literature Review
(OURIQUES et al., 2019)	2019	32	Knowledge Management Strategies and Processes in Agile Software Development: A Systematic Literature Review
(HAFIDZ; SENSUSE, 2019)	2019	15	A systematic literature review of improved knowledge management in agile software development

Regarding the scope of each study, although all studies investigate KM and ASD, each study has a different purpose. In Paredes et al. (2014), the purpose is to summarize information visualization techniques used during the design and development steps of the software cycle by agile teams. Paredes et al. (2014) suggest that visualization techniques are important in ASD because they lead to understanding, collaboration, and self-organization. Visualization techniques help Agile teams to increase knowledge sharing when designing, developing, communicating, and tracking progress.

Cabral et al. (2014) investigated the major KM concepts and findings in ASD projects. The study focuses mainly on main outstanding issues in the software development cycle when tacit knowledge use is prioritized over explicit knowledge. Key findings were discussed in relation to the perspective of a paradigm shift that prioritizes the use of tacit knowledge over explicit knowledge.

Andriyani et al. (2017) conducted a systematic review looking for specific agile practices to support KM, the inherent knowledge involved in these agile practices and how the agile teams manage that knowledge. According to the conclusion, the most important contribution the review provided was an understanding of KM in ASD that involves managing three different types of knowledge - process, project and product - by implementing three KM strategies - discussions, artifacts and visualizations - during agile practices.

The main purpose in Borrego et al. (2017) is architectural KM in Agile and Global Software Development (AGSD). A mapping study was conducted where were identified and described approaches to manage architectural knowledge in AGSD teams. These approaches were grouped as documentation artifact-based, communication-based, and methodological-based.

Neves et al. (2011) analyzed and evaluated the knowledge creation and sharing experiences of teams in the ASD domain. A method was developed to evaluate the advantages and limitations of agile practices in knowledge creation and sharing for agile teams. Considering SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis, assessed agile processes, their relationships with knowledge transfer management and their effects on the productivity of software development teams were assessed.

In Indumini e Vasanthapriyan (2018), a mapping study was conducted to identify existing researches on KM initiatives in ASD in order to present the state-of-the-art in the area. Indumini e Vasanthapriyan (2018) summarize some understandings from mapping: (i) the major problem in organizations are low reuse rate of knowledge and barriers in knowledge transfer in ASD is a recent research; (ii) Knowledge types which are used in ASD are not identified correctly in the organizations; (iii) reuse of development knowledge is the main purpose of applying KM in ASD; (iv) there is a great concern with explicit knowledge using in ASD; and (v) advanced technologies are used to provide KM in ASD.

Hafidz e Sensuse (2019) conducted a Systematic Literature Review to understand more about research development in the KM improvisation in ASD by collecting various themes of improved area and method used. The research question investigated in Hafidz e Sensuse (2019) are “Why improvement in agile still needed?” and “What are the themes of improvement in agile KM proposed by previous research?”. The reasons of the need for agile improvement are: artifact agile lack of documentation detail and traceability; communication and information dissemination problems in the organization; and the increasing of agile process complexion so it needs more supporting tools. Several improvements has been done in different categories such as agile documentation, requirement, process, tools, estimation and decision making. It was possible to identify that many problems in agile already had solutions using different approaches. But there are still some gaps to be improved.

Finally, in Ouriques et al. (2019), a systematic literature review was conducted. A narrative synthesis was applied in order to analyze the data. It was possible to illustrate the KM strategies in the hierarchical layers of software development companies that adopt ASD, through a conceptual classification framework, and provide comprehension concerning how the companies implement KM strategies with practices that promote the knowledge processes.

Once deeply analyzed each paper we started to identify the key concepts addressed in each individual study, making it possible to determine how these concepts are related. This relationship is shown in the next stage of synthesis method.

3.4 PHASE 4 - DETERMINING HOW THE STUDIES ARE RELATED

In the interest for a better understanding of the concepts found in the selected studies, we intend to answer the following RQ in this stage: *What is the study concept presented according to KM and Agile perspectives?* In order to answer this question, we created Table 3 to present the perspectives and their concepts, together with the studies and concepts they have employed.

In relation to KM perspective, knowledge sharing is addressed by all studies. In Andriyani et al. (2017) and Paredes et al. (2014), for instance, one of the concepts is to promote the use of visualization techniques to improve the sharing and retention of knowledge. Agile principles promote producing less documentation. This fact leads to explore strategies of information representation in a leaner and more efficient way, as is the case of Andriyani et al. (2017) and Paredes et al. (2014). In KM, one of the main problems is how to represent knowledge so that it is easily shared (SOUZA, 2014). Visualization techniques can minimize ambiguity and imprecision in interpreting shared information.

In the agile perspective, the two most investigated concepts were Software Artifacts and Communication Team. In Cabral et al. (2014), for example, one of the study classifications is about problems and specific aspects of project documentation in agile methods. In agile projects face-to-face communication is prioritized, instead of the use of documentation. However, some primary studies returned in Cabral et al. (2014) suggest that even though the project is agile, if the team is distributed, there should be a greater focus on knowledge sharing through documents.

KM strategies were investigated by (ANDRIYANI et al., 2017) (included in our KM perspective). According to (ANDRIYANI et al., 2017), with respect to this concept, agile teams use: discussions (e.g. sharing requirements), artifacts (e.g. user stories) and visualizations (e.g. burn-down charts), to manage knowledge. The discussions, in particular, is a verbal communication that involves interaction among agile team members willing to share knowledge. So, in the agile perspective we included the (ANDRIYANI et al., 2017)'s work in "Communication

Table 3: Concept of the studies according to the KM and Agile perspectives

Perspectives	Concepts	Studies
KM	Sharing	(NEVES et al., 2011), (CABRAL et al., 2014), (PAREDES et al., 2014), (ANDRIYANI et al., 2017), (BORREGO et al., 2017), (INDUMINI; VASANTHAPRIYAN, 2018), (OURIQUES et al., 2019), (HAFIDZ; SENSUSE, 2019)
	Capture/Creation	(NEVES et al., 2011), (HAFIDZ; SENSUSE, 2019)
	Knowledge Types	(CABRAL et al., 2014), (ANDRIYANI et al., 2017), (OURIQUES et al., 2019), (HAFIDZ; SENSUSE, 2019)
	Reuse	(CABRAL et al., 2014), (INDUMINI; VASANTHAPRIYAN, 2018)
	Cycle (all activities)	(CABRAL et al., 2014), (PAREDES et al., 2014), (ANDRIYANI et al., 2017), (OURIQUES et al., 2019), (HAFIDZ; SENSUSE, 2019)
	Management Strategies	(ANDRIYANI et al., 2017), (OURIQUES et al., 2019)
	Architectural KM	(BORREGO et al., 2017)
	Knowledge representation	(INDUMINI; VASANTHAPRIYAN, 2018)
Agile	Software Artifacts	(CABRAL et al., 2014), (PAREDES et al., 2014), (ANDRIYANI et al., 2017), (BORREGO et al., 2017), (HAFIDZ; SENSUSE, 2019)
	Communication Team	(NEVES et al., 2011), (CABRAL et al., 2014), (ANDRIYANI et al., 2017), (BORREGO et al., 2017), (OURIQUES et al., 2019), (HAFIDZ; SENSUSE, 2019)
	Collaborative Process	(CABRAL et al., 2014), (OURIQUES et al., 2019), (HAFIDZ; SENSUSE, 2019)
	Agile Practices	(NEVES et al., 2011), (ANDRIYANI et al., 2017), (BORREGO et al., 2017), (INDUMINI; VASANTHAPRIYAN, 2018)
	Architectural Issues	(BORREGO et al., 2017), (INDUMINI; VASANTHAPRIYAN, 2018)
	Technological solutions/-tools	(CABRAL et al., 2014), (BORREGO et al., 2017), (INDUMINI; VASANTHAPRIYAN, 2018), (HAFIDZ; SENSUSE, 2019),

Team”.

Other concepts (purposes) and relationships were identified, as presented in Table 3. This stage helped us to determine how the studies are related. Thus, we started to work on mapping the studies into one another, which corresponds to stage five of the synthesizing method, presented below.

3.5 PHASE 5 - TRANSLATING THE STUDIES INTO ONE ANOTHER

Since our research question relates agile values and agile practices, in this stage we present the relation created from stages 3 and 4. During the relations construction between KM and ASD, we considered the definition of each KM process activity versus the Agile values definition, Scrum events and artifacts, and XP practices presented in the literature. In addition, we used the selected secondary studies in the tertiary study to support our findings. We focused on the three KM activities described in Section 2.1 and the agile values presented in Section 2.3. Table 4 illustrates the relation between KM activities and agile values[1].

Table 4: Relation KM x Agile Values

KM activities	**Agile Values**			
	(I)	**(II)**	**(III)**	**(IV)**
Create/capture and contextualize		*		*
Share, disseminate and assess	*		*	
Acquisition and application		*		*

The study conducted by Neves et al. (2011) presents an analysis and evaluation of KM creation and sharing experiences in ASD teams. Through a specific analysis, the dissertation showed that the values (II) and (III) are directly related to knowledge creation and the values (I) and (III) to knowledge sharing, as illustrates Table 4. In relation to acquisition and application activities, the acquisition of knowledge is the comprehension, amplification and articulation of knowledge in a way that it is internalized. Knowledge application refers to the real use of knowledge that has been captured or created (DALKIR, 2005). Working in the software development or to respond to changes, for instance, experience is required. It is only possible to apply certain knowledge after it had been acquired and the main way to acquire knowledge is through experience. These facts correspond to values (III) and (IV) (CABRAL et al., 2014).

As mentioned earlier (Section 2.3), in order to demonstrate the relation between agile practices and KM activities, we opted to use Scrum and XP frameworks. Besides the eight initial set articles, we also adopted the Scrum Guide (SCHWABER; SUTHERLAND, 2017) and Kent Beck's book (BECK; ANDRES, 2004) as basis for this stage of our study.

[1] The agile values will be referenced by their numbers - "I, II, III and IV", according to the order presented in Section 2.3

3.5.1 TRANSLATING THE SCRUM FRAMEWORK

Table 5 presents the relation among each KM activity and Sprint Events and Scrum Artifacts as well as citations from the eight secondary studies included in the tertiary study that induce or confirm the respective relation presented. A more detailed version of Table 5 is available online on *https://bit.ly/2nD4FNU* (first tab/sheet of the spreadsheet).

There are four columns in Table 5. The first one presents in each column one Scrum activity or artifact. The second one presents a brief description of the relationship between ASD and KM of each Scrum activity or artifact. This information was extracted from Scrum Guide (SCHWABER; SUTHERLAND, 2017). The third and the fourth columns present specific citations (citations that directly support the relationship between one specific Scrum activity or artifact and KM activity) and general citations (citations that generally support the relationship between Scrum activities & artifacts and the KM activity) from our set of selected secondary studies.

Table 5 : Scrum Events and Artifacts x KM Activities

SCRUM	Relation ASD and KM from SCRUM Guide	Specific Citations	General Citations
KM Activity: Create/Capture and contextualize			
Sprint Planning	The work to be performed in Sprint is planned by collaborative work of the entire team. New knowledge can be generated through discussion of the backlog items.	–	"...daily meetings and other activities of interaction among stakeholders are common. Knowledge is potentially stored in people's memory..." (CABRAL et al., 2014).
Daily Scrum	Development team improves communication, eliminate other meetings, identify and remove development impediments, highlight and promote rapid decision making and improve the knowledge level.	"Face-to-face communication in daily meetings (SCRUM), ... (NEVES et al., 2011). ...in a daily stand-up meeting agile teams clarify their cumulative work done by the team..." (ANDRIYANI et al., 2017)	
Development work	During the Development Work, development team members create and build increments.	–	

Product Backlog	Product Backlog is an ordered list of everything that is known to be necessary in the product. It is the unique source of the requirements for any changes to be made to the product.	"... such as product backlog... ....were collectively classified as artifacts since they contained useful knowledge about the software requirements... The product backlog also helped capture product knowledge..." (NEVES et al., 2011)	
Sprint Backlog	Team modifies the Sprint Backlog (plan) during the Sprint. Backlog items are discussed and negotiated with the Product Owner.	–	
KM Activity: Share, disseminate and assess			
Daily Scrum	Team answer question and discuss what was done what will be done and any impediment to achieving the sprint goal.	"...daily meetings with the team (daily scrum) so that knowledge is shared inside the team" (CABRAL et al., 2014).	"...agile methodologies, such as SCRUM, improves organizational learning. ...increase tacit knowledge exchange...". "Verbal communication
Sprint Review	Development team discusses what went well, what problems occurred and how these issues were resolved during Sprint.	–	
Sprint Retrospective	It is made an evaluation of how the sprint was conducted in order to identified improvements to next Sprint.	"In a sprint retrospective project knowledge is shared..." (ANDRIYANI et al., 2017)	

Development work	Development team implements functionality and technology. So, as Scrum promotes collaboration, during development work developers communicate whenever it is necessary.	“Agile teams to increase knowledge sharing when designing, developing...” (PAREDES et al., 2014) “Verbal communication that involves interaction among agile team members which aims to share knowledge...” (ANDRIYANI et al., 2017)	“that involves interaction among agile team members to share knowledge...” (CABRAL et al., 2014) ...knowledge was shared during discussions... sprint review, testing, and small releases...” (ANDRIYANI et al., 2017)
Product Backlog	Product Owner discusses the Product Backlog as it stands, and it is evaluated at each increment delivery.	“Artifacts in agile practices were commonly used to share product knowledge... (e.g. in the form product backlog).” (ANDRIYANI et al., 2017)	
Sprint Backlog	The Sprint Backlog is disseminated and discussed during the sprint and at the end of the sprint, it becomes an increment.	–	
Increment	The increment is shared (presented), reviewed and assessed in the Sprint Review.	–	
KM Activity: Acquisition and application			

Development work	New knowledge is acquired and others are reinforced. The development teams have all necessary skills as a team to create the product increment.	–	–
Increment	An Increment is a body of inspectable, done work that supports empiricism at the end of the sprint. The increment is a step towards a vision or a goal.	“Agile methods provide deliverables after each iteration, ..., thereby facilitating interaction, trust and understanding between on-site customers and the developers” (NEVES et al., 2011)	

Analyzing some relationships identified from Table 5, it is possible to say that the sharing, dissemination and access activities are included in all Scrum artifacts and in almost all Scrum events. Only in the Sprint Planning is not directly identified by literature considered. However, during this Scrum event there is indirect information sharing and dissemination between the PO and the scrum team when they are discussing the Sprint purpose and the Product Backlog items that if completed, they will reach the Sprint goal (SCHWABER; SUTHERLAND, 2017).

Event related to development is not explored in details in the Scrum Guide (SCHWABER; SUTHERLAND, 2017). It is mentioned, but no details are reported on the aspects of its practice. During the research and reflection about the relation between the KM activities and Scrum events, we could notice that the development work is present in all three activities of KM (Table 5). This fact is justified by Scrum values that construct software in a collaborative way.

Few Scrum artifacts and events are related to acquisition and application activities within the context of KM activity. We have identified only one event (development work) and one artifact (increment) that are directly related to this KM activity. This fact induces that more research on the acquisition and application KM activity and ASD should be explored.

3.5.2 TRANSLATING THE XP METHOD

Table 6 presents the relation among each KM activity and XP practices. As same we did for Sprint Events and Artifacts of Scrum, we included citations from the eight secondary studies included in our tertiary study that induce or confirm the respective relation presented. A more detailed version of Table 6 is available online on *https://bit.ly/2nD4FNU* (second tab/sheet of the spreadsheet).

There are four columns in Table 6. The first one presents in each line one XP practice. The second one presents a brief description of the relationship between XP practices and KM activities extracted from Kent Beck's book (BECK; ANDRES, 2004). The third and the fourth columns present specific citations (citations that directly support the relationship between one specific XP practice and KM activity) and general citations (citations that generally support the relationship between XP practices and one KM activity) from the set of selected secondary studies.

Table 6 : XP Practices x KM Activities

XP	Relation between XP practices and KM from Kent Beck's book	Specific Citations	General Citations
KM Activity: Create/Capture and contextualize			
Stories	Stories serve to create the ideas, to place them in a context from which they can be more easily understood by listeners.	–	"They use the knowledge creation model proposed by Tuomi to analyze XP practices and state that these practices enhance knowledge creation..." (CABRAL et al., 2014). "...the interaction during the meetings promotes discussions... This practice allows team members to share knowledge..." (OURIQUES et al., 2019)
Weekly Cycle	To plan work a week at a time in order to review progress to date, pick a week's worth of stories to implement this week, and break the stories into tasks.	"...customer involvement throughout the development cycles allow team members to discuss issues and create knowledge through continuous feedback."(OURIQUES et al., 2019)	
Quarterly Cycle	Refers to a bigger work plan. Once a quarter reflect on the team, the project, its progress, and its alignment. Interval for interaction with external suppliers and customer. In this cycle, it is possible to propose and evaluate long-running experiments quarterly.	"...customer involvement throughout the development cycles allow team members to discuss issues and create knowledge through continuous feedback." (OURIQUES et al., 2019)	
KM Activity: Share, disseminate and assess			

Sit Together	Sit together consists on the practice of having the team members close one each other encouraging conversation.	“...work together for long periods in a distributed context, facilitates knowledge transfer...” (OURIQUES et al., 2019) “...physical workspace affects knowledge transfer effectiveness.” (OURIQUES et al., 2019)	“Walls that separate offices act as barriers for knowledge transfer. ...workspaces that integrate people and furniture that facilitate working together... provide closeness between members.” (OURIQUES et al., 2019) “Process knowledge was shared during discussions...” (ANDRIYANI et al., 2017)
Whole Team	Intense interactions among the team members are necessary for the health of the project. People need a sense of “team”: we belong, we are in this together, we support each others’ work, growth, and learning. What constitutes a “whole team” is dynamic.	“...the members to understand that they need to trust each other and share knowledge... This practice is pointed out... as valuable for knowledge sharing. (OURIQUES et al., 2019)"“...groupware tools to enable frequent communication and build social ties. ...the team’s social potential to share knowledge."(OURIQUES et al., 2019)	

Pair Programming	Pair programming deals with software development and two people who program simultaneously.The developers sit side-by-side and dialog between them.	“The pair programming technique as a form of sharing tacit knowledge...” (CABRAL et al., 2014) “...knowledge exchange efectiveness in pair programming.” (CABRAL et al., 2014) “...most primary studies pay more attention to practices such as pair programming... ...these practices emphasize tacit knowledge sharing..."(ANDRIYANI et al., 2017) “...team members to share their knowledge."(OURIQUES et al., 2019)
Weekly Cycle	To plan work a week at a time in order to review progress to date, pick a week’s worth of stories to implement this week, and break the stories into tasks.	“...intensive communication facilitates knowledge transfer...” (OURIQUES et al., 2019)
Quarterly Cycle	Refers to a bigger work plan. Once a quarter reflect on the team, the project, its progress, and its alignment. Interval for interaction with external suppliers and customer. In this cycle, it is possible to propose and evaluate long-running experiments quarterly.	–

Test-First programming	Test-First programming consist in write a failing automated test before changing any code. It contributes to: delimitation of the scope of the code, focusing objectively and explicitly on what the functionality should do; coupling and cohesion since loosely coupled, highly cohesive code is easy to test; confidence in the quality of the code; development rhythm (test, code and refactor).	–	
KM Activity: Acquisition and application			
Informative Workspace	Consists in make your workspace about your work. The workspace needs to be geared to the projec.	–	–
Continuous Integration	Integration all programmed code and test.	–	
Incremental design	Suggests that the most effective time to design is in the light of experience. In other words, the XP team should make the design of the system an excellent fit for the needs of the system that day.	“Agile methods provide deliverables after each iteration, ..., facilitating interaction, trust and understanding between on-site customers and the developers” (NEVES et al., 2011)	

Regarding to the relationships identified and presented in Table 6, it is possible to state that the sharing, dissemination and access activities are present in most practices of XP (6 out 10 practices considered in Table 6). XP practice that has the strongest relation with the activities of sharing, dissemination and access is pair programming as it is considered the main XP practice and the most cited by the researchers who study KM and XP (practice with more specific citations found – See Table 6 column "Specific Citations").

XP practices "Weekly Cycle" and "Quarterly Cycle" are directly related to the activities of sharing, dissemination and access as well as create/capture and contextualize. This behavior was observed only in these two practices as both practices address development cycles, including planning meetings, project development discussions and other stakeholders (developers, testers, clients, etc.) interactions enabling create/capture and contextualize knowledge and sharing, dissemination and also access knowledge.

Likewise create/capture and contextualize activities, there are only three practices related to acquisition and application activities. This fact reinforces that these both KM activities are not well explored in the XP context yet.

3.6 PHASE 6 - SYNTHESIZING TRANSLATIONS

According to meta-ethnography method (NOBLIT; HARE, 1988), in phase six, a translation synthesis is constructed. Thus, were created two relation maps that summarize the synthesis identified for Scrum and XP in relation to KM. The relation maps are presented in Figure 4 and 5.

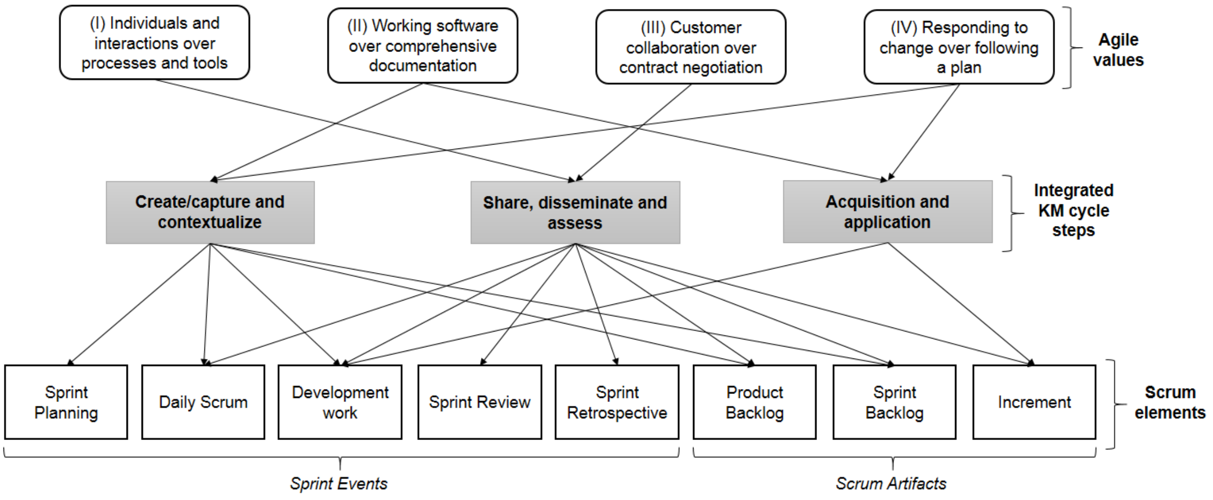
(I) Individuals and interactions over processes and tools
(II) Working software over comprehensive documentation
(III) Customer collaboration over contract negotiation
(IV) Responding to change over following a plan
Agile values
Create/capture and contextualize
Share, disseminate and assess
Acquisition and application
Integrated KM cycle steps
Sprint Planning
Daily Scrum
Development work
Sprint Review
Sprint Retrospective
Product Backlog
Sprint Backlog
Increment
Scrum elements
Sprint Events
Scrum Artifacts

Figure 4: Synthesis of Translations: Relation Map of the relationships between KM, agile values and scrum elements (events and artifacts)

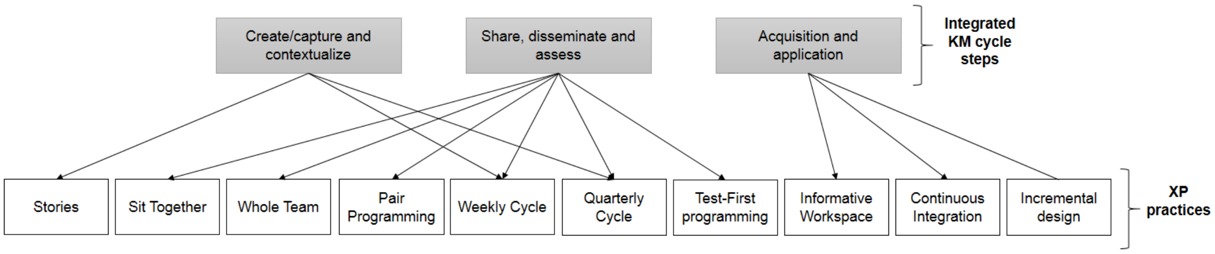

Figure 5: Synthesis of Translations: Relation Map of the relationships between KM and XP practices

In Figure 4 and 5 we evidenced all the concepts and their relationships identified in this research. Figure 4 presents the synthesis of translations in relation to agile values, events and scrum artifacts, and KM activities. Figure 5 presents the synthesis of translations between XP practices and KM activities.

Once this synthesis of translations is created, we decided to validate it by conducting interviews with domain experts. The validation phase is not part of meta-ethnography method. However, we consider this process of validation important to ensure the validity of the concepts and relations between ASD and KM identified, since the synthesis of translations was created from specifics literature, but we also deem it important to have a validation from a practical view of professionals who have been working for many years with agile development.

We conducted a semi-structured interview with four different ASD professionals. Semi-structured interviews allow for improvisation and exploration of the studied objects (WOHLIN et al., 2012). The interviews were conducted at two different times: The agile values events and scrum artifacts interviews occurred in June 2018 with three ASD professionals and the XP practices interviews in May 2019 with two ASD professionals already interviewed at the previous time and a different one. Interviewees were chosen based on the deep experience with ASD and on their availability and willingness to participate in the research. Thus, for the XP practices interviews we changed one interviewee due to his unavailability to participate in the research. Experience time was an important factor in deciding who would be the invited professionals to be interviewed. The interviewees have on average 12 years of experience with ASD, focusing on Scrum and XP frameworks, working in large companies in the South and Southeast regions of Brazil.

The interview questions were created considering each relationship in our synthesis of translations presented in Figure 4 and 5. For each relationship we create a question in the affirmation format, for example, for the relationship between the concepts "**Create/capture and contextualize**" and "**Sprint Planning**" (see Figure 4) the following statement was created: "*In* *agile methods, the Sprint Planning event is the time when the team meets to plan and organize the work to be done. It is at that moment that the internal knowledge is identified and created, that is, at this moment new knowledge can arise. In this activity, the group tacit knowledge is captured.*" In order to answer each affirmation, the interviewee chose a degree of agreement considering a scale of 1 to 5 based on the Likert Scale method (LIKERT, 1932).

For each affirmation closed question presented earlier, we have created an open question with the intention of identifying examples or reports of the respondent's life experience in relation to their response. In this moment, new questions spontaneously have arisen, which allowed for a better conversation flow and better information.

All interviews were conducted through Skype or Hangouts video conference and lasted approximately one hour each. Their answers were transcribed for analysis and interpretation. The whole interviews questions and the answers format can be accessed at Appendix A.

In relation to the relationships validation for the agile values, events and scrum artifacts, and KM activities created in Figure 4, 14 relationships were confirmed from the three interviewees answers with scales equaling to 5; this represents 82.3 % of the synthesis of translations relationships for for the agile values, events and scrum artifacts. Out of the other three questions in which there was no consensus on full agreement, two interviewees pointed out a conformance on scale 4, which is a strong indication of the relationships created. However, in one of the questions the interviewee answered a scale 2 for agreement. Basically the question stated that in Sprint Backlog new explicit knowledge is generated and organized ("**Create/capture and contextualize**" x "**Sprint Backlog**"). We asked the interviewee why he did not fully agree with the statement, and the answer was that in his opinion the most of the knowledge is generated in the sprint planning process. In sprint planning the tasks are discussed and explained, but in the sprint backlog, although there may be increases in knowledge, it is little relevant.

The agreement of two area experts with the "**Create/capture and contextualize**" x "**Sprint Backlog**" relationship still indicates the relationship existence. Even so, this item will be discussed in the item of threats to validity.

Regarding to the relationships validation for the XP practices and KM activities created in Figure 5, 10 from the 12 relationships are confirmed by the following response pattern: two interviewees answers with scales equaling to 5 and one interviewee answers with scales equaling to 4. This pattern represents 83.3% of the synthesis of translations relationships for the XP and it means that the majority of the XP practices are strongly connected with KM activities.

In the question stated that the practice “Sit Together” allows sharing and exchange of knowledge among the team members (“**Share, disseminate and assess**” x “**Sit Together**”), two interviewees pointed out a conformance on scale 4 and one interviewee on scale 5. Both who pointed out 4 mentioned in the comments that the practice of sitting together is not always productive, the practice needs to be implemented carefully in order to keep all involved people focused and productive.

All interviewees have already experienced the practice of pair programming by themselves and with their teams. Regarding the question about the relation between “**Share, disseminate and assess**” and “**Pair programming**” two interviewees fully agreed with this relationship and mentioned in the comments that during the pair programming “there is an intense exchange of knowledge through effective communication and collaborative work” and that this practice “collaborate with team knowledge”. However, one interviewee pointed out a conformance on scale 3 mentioning in the comments a relevant remind: “I believe that practicing pair programming 100% of the time loses productivity. It is important for the team to be trained and integrated to avoid that knowledge stay only with part of the team, consequently, the knowledge is not shared and disseminated properly”.

The interviewees also made important positive notes and reports of the lived experiences. In summary, all the interviewees emphasize, in the open questions, that the creation, transfer or retention of knowledge within the companies that work with ASD are of great importance. For them software is knowledge transformed into code, so it is fundamental to take care of this information and keep it inside the company. One interviewee emphasized that creation, transfer or retention of knowledge in an ASD organization helps to reduce the high costs with employee changes. The experts mentioned that companies want to be agile, but they do not understand the effort in relation to the change that needs to happen. Companies are very much attached to contract, scope and deadline, but what the companies should do is deliver the value (what the customer needs).

3.7 PHASE 7 - EXPRESSING THE SYNTHESIS

In this last stage the objective to report the synthesis result. According to Silva et al. (2013a), the target audience is the research community interested in performing synthesis of empirical research using meta-ethnography. Thus, it is believed that synthesis can help direct researchers in future research by providing guidance to properly position new research activities.

3.8 DISCUSSION OF SYNTHESIS RESULTS

We began this study with the intention of answering the following question "*How do agile values and practices relate to KM activities?*". Meta-ethnography method was used to identify the existing synthetics between these areas. As pointed out by (SILVA et al., 2013a), this method requires several readings of the initial set of studies, data extraction, consensus meeting, reaching agreements and verifying potential inconsistencies in interpretations, time consuming and requires maturity in the research team. However, although the method is not simple to apply, we believe that we have been able to answer the research question by means of a synthesis of translations between KM, agile values, Scrum elements (events and artifacts) and XP practices considered plausible of comprehension. This synthesis of translations created can provide a contribution mainly to researchers and interested in understanding the relationships in this domain: KM and ASD.

A synthesis study focuses primarily on primary studies. However, we realized that the method could also be applied in secondary studies returned from a tertiary study. And even working with eight articles in the initial set, when necessary we consulted some of 238 primary studies returned in our eight secondary studies.

In relation to the study results, we can say that sharing is a KM activity that is strongly related to ASD and very investigated in Scrum and XP frameworks. This makes sense since ASD prioritizes the exchange of information and communication among teams. Agile values promote a focus on the people involved in a project and how they interact and communicate. The communication is strongly related to KM, and in agile teams the focus is the tacit knowledge. Also noteworthy that creating knowledge is a strongly present activity in Scrum, but not to the same degree in XP.

We also note that the activity of acquisition and application has not been investigated like the other activities from both Scrum and XP. Knowledge applying activity is the last step of the KM cycle which suggests that the captured and shared knowledge must ultimately be used in the organization. So, the acquisition and application are activities that could be better explored in both industry and academia.

4 CONCLUSIONS

4.1 GENERAL CONSIDERATIONS

In this dissertation, we reported on the results from an analysis on KM and ASD methodology using the meta-ethnography method. We applied the seven phases of meta-ethnography analysis method on eight articles selected from a tertiary study on KM and ASD. After, the syntheses identified in these areas investigated were analyzed based on interviews with three ASD consultants.

Agile practices and KM present common activities that can encourage software organizations to promote KM activities. The exploration of KM activities during ASD can improve team learning and collaborate with the evolution of organizational knowledge leading to deliveries with greater value and consequently increase the customer satisfaction. The most common activity between these two areas is knowledge sharing. In (CABRAL et al., 2014), they presented that there is still a gap about what emerges from the intersection of these two areas that needs further clarification. In our investigation, we believe that the acquisition and application activities are research topics that can be explored more.

4.2 MAIN CONTRIBUTIONS

The major contribution of this study was to provide the understanding of how specific dimensions of KM in ASD can help area practitioners effectively manage the knowledge in everyday agile actions. Several organizations that adopt agile methodologies face a problem related to the management and organizational knowledge retention. The agile premise is guided by contributory and collaborative work as well as knowledge evolution. The clarification of how KM is present in each agile value, Scrum event and artifacts allows a reflection on how much knowledge has been created, shared and applied during ASD. This reflection enables organizations explore more each KM cycle phase, consequently, contributing to a delivery with greater value to the client.

In addition, the conduction of the meta-ethnography method can also be considered a contribution. According to Fu et al. (2019), there is a need of investing in gaining knowledge and experience in applying meta-ethnography. High quality meta-ethnography can make significant contributions to empirical SE and it is believed that this dissertation presents a new example of how to relate two different areas and how to apply the method in practice. So, this dissertation can also support the improvement of the use of meta-ethnography in SE as well as the advance of future empirical research in this context.

During the development of this project, some papers were published/submitted:

RUIZ, G. A.; NAPOLEÃO, B. M.; DE SOUZA, E. F.; FELIZARDO, K. R.; MEINERZ, G. V.; DA SILVA, P. R.; VIJAYKUMAR, N. L. **Using meta-ethnography to synthesize research on knowledge management and agile software development methodology.** 17th Brazilian Symposium on Software Quality (SBQS), p. 230-239, 2018.

RUIZ, G. A.; DA SILVA, P. R.; SOUZA, E. F.; VIJAYKUMAR, N. L.; FELIZARDO, K. R.; MEINERZ, G. V. **Knowledge Management in Agile Testing Teams: a Survey.** Experimental Software Engineering Latin American Workshop (ESELAW'18), XXI Ibero-American Conference on Software Engineering (CIBSE)-Experimental Software Engineering (ESELAW) Track, Bogotá, Colombia, 2018.

NAPOLEÃO, B. M.; RUIZ, G. A.; DE SOUZA, E. F.; FELIZARDO, K. R.; MEINERZ, G. V.; VIJAYKUMAR, N. L. **Synthesizing researches on Knowledge Management and Agile Software Development using the Meta-ethnography Method.** Paper submitted to Journal of Systems and Software.

4.3 LIMITATIONS

One threat to validity in this study could be the research method used. Since meta-ethnography is an interpretive approach to synthesis, we addressed validity and reliability of our synthesis from an interview with three ASD experts from a practical view.

A larger number of interviews could be consulted and improve this research quality; however, given the interviewees profile, we believe that a single interview could already bring important observations to the research. Even so, we intend to conduct a survey in order to continues validating our synthesis.

Some relationships between KM, agile values and practices may not have appeared in our relation map. This is due to the fact that we considered only eight secondary studies, Scrum guide and Kent Beck's book. An alternative to this limitation, would be to increase the set of references considered during the relationships construction and to consider a greater number of interviewees in the validation process.

4.4 FUTURE WORKS

A richer investigation with better mechanisms to perceive KM in ASD forms part of future work. We intend to complement our research with other empirical methods, for example, we intend to conduct a survey in software organizations and to extend this analysis adopting others agile methodologies.

REFERENCES

ANDRIYANI, Y.; HODA, R.; AMOR, R. Understanding knowledge management in agile software development practice. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE SCIENCE, ENGINEERING AND MANAGEMENT (KSEM). Melbourne, VIC, Australia, 2017. p. 195–207.

BECK, K.; ANDRES, C. **Extreme Programming Explained: Embrace Change**: Addison-Wesley Professional, 2004. ISBN 0321278658.

BECK, K. et al. **Manifesto for Agile Software Development**. 2001. “http://www.agilemanifesto.org/”.

BJØRNSON, F. O.; DINGSØYR, T. Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. **Information and Software Technology**, v. 50, p. 1055–1068, 2008.

BORREGO, G. et al. Review of approaches to manage architectural knowledge in agile global software development. **IET Software**, v. 11, n. 3, p. 77–88, 2017.

BUKOWITZ, W.; WILLIAMS, R. L. **The knowledge management fieldbook**. Great Britain: Financial Times Prentice Hall, 2000.

CABRAL, A.; RIBEIRO, M.; NOLL, R. Knowledge management in agile software projects: A systematic review. **Journal of Information & Knowledge Management**, v. 13, p. 1–37, 2014.

COBB, C. **The Project Manager’s Guide to Mastering Agile: Principles and Practices for an Adaptive Approach**: Wiley, 2015. (Business et Economics : Project Management). ISBN 9781118991046.

CRUZES, D. S.; DYBå, T. Recommended steps for thematic synthesis in software engineering. In: **In Empirical Software Engineering and Measurement (ESEM)**, 2011. p. 275–284.

CRUZES, D. S.; DYBå, T. Research synthesis in software engineering: A tertiary study. **Information and Software Technology**, v. 53, n. 5, p. 440–455, 2011.

DALKIR, K. **Knowledge management in theory and practice**: Routledge, 2005.

DAVENPORT, T. H.; PRUSAK, L. **Working knowledge**: how organizations manage what they know. 2. ed. Boston, USA: Harward Business School Press, 2000.

DYBA, T.; DINGSOYR, T. Empirical studies of agile software development: A systematic review. **Information and Software Technology**, v. 50, p. 833–859, 2008.

FELIZARDO, K. R. et al. **Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática**: Elsevier Brasil, 2017.

FU, C. et al. A review of meta-ethnographies in software engineering. In: **Evaluation and Assessment in Software Engineering (EASE'19)**, 2019.

HAFIDZ, M. U. A.; SENSUSE, D. I. A systematic literature review of improved knowledge management in agile software development. In: **2Nd International Conference on Software Engineering and Information Management**New York, NY, USA: , 2019. (ICSIM 2019), p. 102–105. ISBN 978-1-4503-6642-7.

HANNES, K.; LOCKWOOD, C. **Synthesizing qualitative research: Choosing the right approach**: John Wiley & Sons, 2011.

INDUMINI, U.; VASANTHAPRIYAN, S. Knowledge management in agile software development - a literature review. In: **National Information Technology Conference (NITC)**Colombo, Sri Lanka: , 2018.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing Systematic Literature Reviews in Software Engineering**, 2007.

LEVY, M.; HAZZAN, O. The knowledge life cycle. In: **ICM Conference on KM**Miami, FL: , 1999.

LEVY, M.; HAZZAN, O. Knowledge management in practice: The case of agile software development. In: ICSE WORKSHOP ON COOPERATIVE AND HUMAN ASPECTS ON SOFTWARE ENGINEERING (ICSE CHASE'09). Vancouver, BC, Canada, 2009. p. 60–65.

LIKERT, R. A technique for the measurement of attitudes. **Archives of psychology**, 1932.

MAPLESDEN, D. et al. Performance analysis for object-oriented software: A systematic mapping. **IEEE Transactions on Software Engineering**, IEEE, v. 41, n. 7, p. 691–710, 2015.

MEYER, M.; ZACK, M. The design and implementation of information products. **Sage Publications Inc**, 1996.

NEVES, F. T. et al. Knowledge creation and sharing in software development teams using agile methodologies: Key insights affecting their adoption. In: **6th Iberian Conference on Information Systems and Technologies (CISTI)**, 2011. p. 1–6.

NOBLIT, G.; HARE, R. **Meta-Ethnography**: Synthesizing qualitative studies (qualitative research methods). Newbury Park, Calif.: Sage Publications Inc, 1988.

NONAKA, I.; TAKEUCHI, H. **The knowledge creation company: how Japanese companies create the dynamics of innovation**: New York: Oxford University Press, 1995.

O'LEARY, D.; STUDER, R. Knowledge management: an interdisciplinary approach. **IEEE Intelligent Systems**, v. 16, 2001.

O'LEARY, D. E. Enterprise knowledge management. **IEEE Computer Magazine**, p. 54–61, 1998.

OURIQUES, R. A. B. et al. Knowledge management strategies and processes in agile software development: A systematic literature review. In: **International Journal of Software Engineeering and Knowledge Engineering**, 2019. p. 345–380.

PAREDES, J.; ANSLOW, C.; MAURER, F. Information visualization for agile software development teams. In: IEEE WORKING CONFERENCE ON SOFTWARE VISUALIZATION, 2014. p. 157–166.

PRESSMAN, R. S. **Engenharia de Software. Tradução: Rosângela Delloso Penteado**: São Paulo: McGraw-Hill, 2006.

RUIZ, G. A. et al. Using meta-ethnography to synthesize research on knowledge management and agile software development methodology. In: **17^{th} Brazilian Symposium on Software Quality (SBQS)**Curitiba, Paraná, Brazil: , 2018. p. 230–239.

SCHÖN, E. M.; ESCALONA, M.; THOMASCHEWSKI, J. Agile values and their implementation in practice. **Journal of Interactive Multimedia and Artificial Intelligence**, 2015.

SCHWABER, K.; SUTHERLAND, J. **The Scrum Guide**. https://www.scrumguides.org/download.html, 2017.

SILVA, F. et al. Using meta-ethnography to synthesize research: A worked example of the relations between personality and software team processes. In: **7^{th} International Symposium on Empirical Software Engineering and Measurement (ESEM)**, 2013. p. 153–162.

SILVA, F. Q. B. D. et al. Using meta-ethnography to synthesize research: A worked example of the relations between personality and software team processes. In: IEEE. **Empirical Software Engineering and Measurement, 2013 ACM/IEEE International Symposium on**, 2013. p. 153–162.

SOUZA, E. F. **Knowledge management applied to software testing: an ontology based framework**. Tese (Doutorado) — Instituto Nacional de Pesquisas Espaciais (INPE), Brazil, 2014.

SOUZA, E. F.; FALBO, R. A.; VIJAYKUMAR, N. L. Knowledge management initiatives in software testing: A mapping study. **Information and Software Technology**, v. 57, p. 378–391, 2015.

SUTHERLAND, J.; SCHWABER, K. **The Scrum Papers: Nut, Bolts, and Origins of an Agile Framework. 224p. 2011**. 2012.

VASANTHAPRIYAN, S.; TIAN, J.; XIANG, J. A survey on knowledge management in software engineering. In: INTERNATIONAL CONFERENCE ON SOFTWARE QUALITY, RELIABILITY AND SECURITY COMPANION (QRS-C). Vancouver, BC, Canada, 2015. p. 237–244.

WENDORFF, P.; APSHVALKA, D. The knowledge management strategy of agile software development. In: 6TH EUROPEAN CONFERENCE ON KNOWLEDGE MANAGEMENT. Univ. of Limerick, Ireland, 1998. p. 607–614.

WIIG, K. **Knowledge management foundations**. Arlington, TX: Schema Press, 1997.

WOHLIN, C. et al. **Experimentation in software engineering**: Springer Science & Business Media, 2012.

APÊNDICE A – INTERVIEWS QUESTIONS

Interview

Contextualization: This questionnaire aims to provide a better understanding of how Knowledge Management activities are related to Agile Practices (Agile values, Scrum framework and eXtreme Programming method). The main activities of Knowledge Management are Knowledge Capture (development of new knowledge that did not previously exist in the organization); Sharing Knowledge (knowledge is shared and disseminated throughout the organization); and Application of Knowledge (the knowledge captured and shared is put in use).

Part I - Interviewee Profile I am aware and agree that my answers will be used for research purposes.

( ) Yes ( ) No

Name:

Company Name:

What is the Agile framework or methodology that you most work?

( ) SCRUM

( ) XP (Extreme Programming)

( ) Crystal

( ) Other. Which one?

What is your position or role in relation to some agile framework or methodology? (i.e. Agile Facilitator, Scrum Master, Product Owner, Agile Coach, Developer, etc.)?

How long do you have experience with agile framework or methodology?

For you, what is the importance of creating, transferring, and retaining knowledge inside a company?

Part II - Relation of Agile Values, Events and Scrum Artifacts with Knowledge

Management

Contextualization: Tacit knowledge is developed through intuition, observation and practice. Because it is not a type of knowledge easy to be formalized and explained, it is usually retransmitted by daily living and contact with the knower, being considered as a differential in one person. It is stored in people’s minds in the form of memory, skills, experience, education, imagination and creativity. Explicit knowledge is communicated and understood through words, images, graphs and methodologies. It is a type of knowledge based on rationality; it is regulated, theoretical and can be learned through texts, books, classes, software products, etc.

This section of the questionnaire is composed of statements that you should put in the answer scale that shows how much you agree with the statement by following these criteria:

1.Strongly disagree

2.Disagree

3.Neutral

4.Agree

5.Strongly agree

1- Within the agile methods, the Sprint Planning event is the moment when the team meets to plan and organize the work to be done. It is at this moment that the internal knowledge is identified and created, that is, at this moment new knowledge can arise. In this activity, the group?s the tacit knowledge is captured.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

2- The Scrum event known as the Daily Scrum promotes a daily meeting with the purpose of describing what was done in the previous day, identifying impediments and prioritizing the work to be done during today. This event allows the creation of knowledge through the participants? exchange of experience (tacit knowledge).

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

3- During Development Work in Scrum, team members perform their daily tasks as defined in the Daily Scrum. At the time of the execution of the tasks, a new knowledge can be captured and later organized to be presented in an explicit format and also be disseminated more easily throughout the organization.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

4- Product Backlog is a Scrum artifact maintained and transformed into increments of functionality potentially releasable in each Sprint. Everything that is known to be necessary in the product should be part of the Product Backlog. Thus, from the creation of the items of the Product Backlog it is possible to affirm that new explicit knowledge is being generated and organized.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

5- The Sprint Backlog is a set of Product Backlog items selected for Sprint with a plan to deliver the product increment and achieve the Sprint purpose. Like the Product Backlog, the Sprint Backlog allows the generation and organization of new explicit knowledge.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

6- In the Daily Scrum it is essential to share knowledge among the team members, this encourages the dissemination of the knowledge of each individual.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

7- During the Sprint Review Meeting the development team presents the increment to all stakeholders and discusses what went well during Sprint, what problems occurred and how these issues were resolved. This characterizes sharing knowledge among those involved.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

8- At the Sprint Retrospective, the team is encouraged to review its development process in order to make it better for the next Sprint. During this type of meeting, knowledge is sharing among the entire team and others people involved.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

9- Product Backlog is a list of desired functionalities of a product, that is, the requirements that a customer expects to receive at the end of the project, described in his own language. To better structure the Product Backlog, user stories are used, which contains a detailed description of the requirements of each request to be implemented. Throughout this process there is sharing of knowledge, because there is a great exchange of information and experiences among all people involved to better understand the customers? needs raised.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

10- For the definition of the items that will be inserted in the Sprint Backlog, the exchange of information and knowledge among the team is fundamental for a good process structuration.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

11- During the Sprint development (Development Work), developers acquire and apply knowledge all the time, performing their tasks, helping other developers and learning constantly.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

12- Agile methods evidence team collaboration, manage project problems and complexities using functionality-based decomposition that is followed by integration of software increments, and finally emphasize the communication of technical details using verbal forms, graphics and text. In this way, it is evident that in the artifact Increment is identified the acquisition and application of knowledge.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

13- One of the agile values describes "Individuals and interactions over processes and tools". This interaction is strongly characterized by sharing and disseminating knowledge within the context of agile methods.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

14- "Working software over comprehensive documentation"is one of the agile values that more demonstrates that during the software design process, there was knowledge creation, capture and contextualization.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

15- Among the agile values, the value "Customer collaboration over contract negotiation"shows that sharing, disseminating and assessing are extremely accomplished steps to obtain a product that adds value to the customer.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

16- "Responding to change over following a plan"is one of the agile values and indicates that every project must balance planning and changing based on its level of uncertainty. In projects where there is high uncertainty, research and design predominate over the planning and the limited execution of planned tasks. Share, disseminate and evaluate is a knowledge cycle used frequently.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

17- One of the agile values that requires acquisition and application of knowledge is "Responding to change over following a plan". Thus, it is clear where companies seeking to prosper in today’s turbulent economy must use knowledge in order to change their processes and their perspectives in relation to changes, since that everything (except the product vision) can change in a very short time, for example: product scope, functionalities, technology, architecture, etc.

○	○	○	○	○
Strongly disagree	Disagree	Neutral	Agree	Strongly agree

Comments:

Part II - Relation of eXtreme Programming (XP) Practices with Knowledge Management

Contextualization: Tacit knowledge is developed through intuition, observation and practice. Because it is not a type of knowledge easy to be formalized and explained, it is usually retransmitted by daily living and contact with the knower, being considered as a differential in one person. It is stored in people’s minds in the form of memory, skills, experience, education, imagination and creativity. Explicit knowledge is communicated and understood through words, images, graphs and methodologies. It is a type of knowledge based on rationality; it is regulated, theoretical and can be learned through texts, books, classes, software products, etc.

This section of the questionnaire is composed of statements that you should put in the answer scale that shows how much you agree with the statement by following these criteria:

1.Strongly disagree

2.Disagree

3.Neutral

4.Agree

5.Strongly agree

1- XP teams plan what needs to be done using stories written on small cards. Developers team use face-to-face dialogue with the client to learn as much as possible about the details of each story. In this way, the record of the story done on the card ends up being used as a reminder of the dialogue. It can be said that the main difference between stories and other requirements practices is that stories allow early estimation. Stories estimation gives businesses and technical prospects a chance to interact, which creates value in advance when an idea has greater potential by focusing on the benefit and the business value to the customer. In this XP practice occurs knowledge creation through the interactions between client and team.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

2- Inside XP, the Sit Together practice is the time when XP team members sit together in an open room where everyone can work together and communicate faster and more efficient. This practice predicts that the more face-to-face the team members are, more productive the project will be. It is in this moment that knowledge is shared among team members, that is, there is the exchange of knowledge between involved people.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

3. Whole Team is an XP practice that is based on the principle that the result of a project depends not only from developers, but also from any person who may have some kind of contribution to the project (i.e. client). Team spirit is very important for the belonging to the team happen (we are on this together; we support each other's work, growth and learning). The idea in this practice is client, and others important people in the project, have (maximum possible) availability to help developers whenever they have questions. This practice empowers knowledge sharing and it is fundamental for a development project succeed.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

4- Pair programming suggests that any code produced in the project is always implemented by two people together, in front of the same computer, taking turns on the keyboard. It is an effective way to reduce the incidence of bugs in a system due to in large of the complementary visions generated during the use of this practice. Developers are working together to identify solutions. In addition to sitting side by side, they talk all the time, reflecting on system refinements, taking initiatives when the partner is "stuck"and more important: exchanging ideas about the solution. Sharing knowledge is evident in this XP practice.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

5- Software is developed iteratively and incrementally in XP projects. Once a week (Weekly Cycle) developers meet with the client to prioritize a small set of features that can be fully implemented and tested in a week. Finished this period, called iteration, the client has the opportunity to use and evaluate what was developed. Based on the results, the client (on his/her representative) meet again with the team and set new priorities according to what was learned from the software and what he/she already thought that needed to produced over the rest of the project. It is possible to affirm that each weekly cycle (Weekly Cycle) knowledge is created and shared among those involved.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

6- The overall planning of an XP project is divided into quarters (Quarterly Cycle). At the beginning of each quarter the team meets with the client and/or others interested in the project to establish the theme(s) that will be implemented over the next three months. Topics are sets of features that address the needs of one or more business processes in an organization. In the planning of a quarterly cycle, the team should also reflect about the project progress from the previous quarter, identify problems and propose actions to solve these problems in the new quarter. During the quarter themes alignment and in the reflection of the project progress is generated knowledge. Also during quarterly planning the knowledge among stakeholders is shared in order to identify and propose actions to solve problems and understand what will be implemented in the next quarter.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

7- Test-driven development consists in developing functionality automated test before developing it. The main objective of this practice is to anticipate the identification and correction of failures during development. This practice contributes to delineating the scope of the code, focusing on what functionality should do; coupling and cohesion since a highly cohesive code is easier to test; increase the confidence and quality of the developed code; and creation of a development rhythm (test - development - refactoring). Constant interaction during testing implementation between developers, clients, and testers characterizes knowledge sharing among them.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

8- An informative workspace of an XP team should be a reflection of the project. They have a work space oriented to the project, a place where someone who enters in the team room can in a few seconds have a clear sense of how the project is going. This practice improves team communication and help them saving-time since the most important information about the project is always visible on the walls. Such information must be active and it must be updated frequently to make them valuable. This practice promotes the acquisition of knowledge.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments:

9- In a multi-developer team with all developers working on the same system, it is possible to have a way of unify the various changes made to the code basis. The practice known as Continuous Integration can solve this issue. Continuous integration consists in integrate all programmed code and testing changes several times a day (the longer you wait to integrate, bigger are the costs), ensuring that the code basis remains consistent at the end of each integration. This practice promotes acquisition and application of knowledge in order to the team make progress and finish their project activities.

Comments:

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

10- In the XP practice called Incremental Design, the design of an application appears iteratively and incrementally in projects. The goal is to create the simplest possible solution that is sufficient to implement the functionality of each iteration. Any feature that can be implemented to support a future functionality is only coded if and when this functionality is prioritized (in a future iteration). Thus, it is aimed to focus the team's efforts on what is absolutely clear that it will be necessary today (since it has already been prioritized by the client for the current iteration). What could be useful in the future, is leaved to resolve in the future when there is clear its necessity. Acquisition and application of knowledge occurs during interactions to enable assertive definitions in the project.

○ Strongly disagree ○ Disagree ○ Neutral ○ Agree ○ Strongly agree

Comments: