Knowledge Management Diagnostics in Software Development Organizations: a Systematic Literature Review

Claudia P. C. Maciel
Department of Computer
Federal Technological University of Paraná – UTFPR
Cornélio Procópio, PR, Brazil
claudiamaciel@alunos.utfpr.edu.br

Érica Ferreira de Souza
Department of Computer
Federal Technological University of Paraná – UTFPR
Cornélio Procópio, PR, Brazil
ericasouza@utfpr.edu.br

Ricardo de Almeia Falbo
Depart. of Computer Science,
Federal University of Espírito Santo – UFES
Vitória, ES, Brazil
falbo@inf.ufes.br

Katia Romero Felizardo
Department of Computer
Federal Technological University of Paraná – UTFPR
Cornélio Procópio, PR, Brazil
katiascannavino@utfpr.edu.br

Nandamudi L. Vijaykumar
Lab. of Computing and Applied Mathematics, National Institute for Space Research – INPE
São José dos Campos, SP, Brazil
vijay.nl@inpe.br

ABSTRACT

Context: Managing knowledge is one of the main challenges for software development organizations. Thus, the principles of Knowledge Management (KM) are presented as determinant and effective factors for the software product quality. There are several approaches to applying KM in an organization. However, for a KM approaches to succeed, it is important to conduct a KM diagnostic in order to analyze the KM current state that already exists in the organization. **Objective:** The objective of this paper is to present the results of a Systematic Literature Review (SLR) conducted to summarize existing research on KM diagnostic in software development organizations. **Method:** SLR was performed by searching four electronic databases. We also performed backward snowballing from reference lists of selected studies. **Results:** From the SLR, we identified 24 studies addressing investigated differents approaches related to KM diagnostics in software development organizations. **Conclusion:** Based on our results we conclude that in the software engineering context the KM diagnosis practice still does not seem consolidated.

CCS CONCEPTS

- **Software and its engineering → Software creation and management**;

SBQS, October 17–19, 2018, Curitiba, Brazil

ACM ISBN 978-1-4503-6565-9/18/10...$15.00
https://doi.org/10.1145/3275245.3275260

KEYWORDS

Software Engineering, Knowledge Management, Knowledge Management Diagnostic

ACM Reference Format:
Claudia P. C. Maciel, Érica Ferreira de Souza, Ricardo de Almeia Falbo, Katia Romero Felizardo, and Nandamudi L. Vijaykumar. 2018. Knowledge Management Diagnostics in Software Development Organizations: a Systematic Literature Review. In *17th Brazilian Symposium on Software Quality (SBQS), October 17–19, 2018, Curitiba, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3275245.3275260

1 INTRODUCTION

Creating systematic strategies to integrate the knowledge in organizations is considered a great challenge [4]. Several researches present different approaches in order to capture and share knowledge in an organization, making knowledge transparent to all those involved [30]. Working with organizacional knowledge makes and amplifies knowledge created by individuals as well as crystallizing and connecting it to an organization's knowledge system [40].

Knowledge Management (KM) has emerged as a discipline with the purpose of acquiring, organizing and sharing the knowledge in the organization, so that other members can use it to make their work more effective and productive. The main goal of KM is to make organizational knowledge accessible and promote the emergence of new knowledge [42]. According to Bukowitz and Williams [5], KM also is the process by which the organization generates wealth from its knowledge or intellectual capital. In summary, KM is the process through which organizations generate value from their intellectual assets.

In relation to software engineering companies, a software development process has as main aspects the rapid changes, the use of sharp knowledge and the involving several people in different activities and roles. Knowledge involved in all these aspects is a fundamental incentive for the use of KM iniciatives [7]. In the last decades, increased consideration was given to the improve software quality considering Maturity Models and ISO standards. However, the need to improve software quality adds to the demand for systematic knowledge [3]. KM incorporation in Software Engineering has been implemented in order to promote the experience of reuse to improve processes and products related to software development, as well as solutions and problems traceability, document management, software reuse, support for project memories and the learning core [4]. The benefits and the need to apply KM in Software Engineering have been reported in several studies [1], [4], [7], [34], [52], [57].

The literature presents different approaches, also known by KM cycles or models, that propose an activities set in order to promote KM initiatives [9]. A KM cycle has activities that encompass, for example, capturing, creating, encoding, sharing, accessing, applying and reusing the individual, group and organizational knowledge within and between organizations. However, while there are several KM approaches, applying KM without first looking at the organization current state may require high and often ineffective investments in the identification and sharing of knowledge that are in fact relevant to the organization [5], [48], [14]. Conducting an organization diagnostic can determine which areas really need improvement or which areas are most cost-effective for organization in terms of KM [5].

Few organizations have time to apply an entire KM cycle, then a diagnosis of how KM approach are being carried can be conducted to identify which areas of KM have the most potential to work with knowledge [5]. Analyzing the organization current state in relation to KM can help the organizations members understand their actual needs before dedicating expensive efforts in the KM implementation and thus better target the KM application initiatives in companies' strategic points.

A clear understanding about how KM diagnostic has been applied in software companies is important to steer future research. However, there is still little knowledge about the state of the art with respect to this research area in Software Engineering. The objective of this paper is to present a Systematic Literature Review (SLR) conducted to summarize existing research on KM diagnostic in software development companies. SLR was performed by searching four electronic databases. We also performed backward snowballing from reference lists of selected studies, in order to identify additional relevant studies. We considered studies published until June 2018.

The remainder of this paper is structured as follows. Section 2 review the literature of KM and diagnostic, as well as related works. Section 3 introduces the methods and procedures used to conduct the research. Section 4 presents the selection process conducted. Section 5 shows the main results from studies extraction and synthesis. Section 6 reports a general discussion to highlight some research points, their implications, and limitations. Lastly, conclusions and future directions for this research are presented Section 7.

2 BACKGROUND

In this section, the main concepts of this study and related works are discussed briefly.

2.1 Knowledge Management and Software Engineering

The creation and dissemination of knowledge within organizations have become ever more important factors in competitiveness. Knowledge has been regarded as a valuable commodity that is embedded in high-technology products and in the knowledge of highly mobile employees [9]. KM is defined as the process of applying a systematic approach to capture, structure, manage, and disseminate knowledge throughout an organization in order to reuse best practices [41, 46]. KM solutions have proven to be most successful in disseminating knowledge that has been rendered from Lessons Learned (LL) and best practices.

Among the knowledge types that can be managed within an organization, one of the most valuable is Tacit Knowledge. Tacit knowledge depends on personal experience and involves intangible factors such as beliefs, perspectives, values and intuition [41]. This type of knowledge cannot fully be articulated, since involves knowledge tied to the senses, skills, experiences or intuition [40]. Another important knowledge type is Explicit Knowledge. Contrary to tacit knowledge, the explicit knowledge represents the objective and rational knowledge that can be uttered and captured in images and writing, and can be easily used and shared.

Several organizations everywhere and in every segment are paying attention to knowledge and are beginning to actively manage their intellectual capital starting to explore what it is and how to create, transfer, and use effectively the knowledge [10]. In software engineering organizations this is no different. Integrating KM in software engineering has become essential. There are many approaches on how software should be developed and which can affect how knowledge is managed. Traditional development, for instance, rely primarily on managing explicit knowledge, and in turn, agile methods, primarily rely on managing tacit knowledge [38, 57]. According to Bjørnson and Dingsøyr [4], in software engineering, there has been much discussion about how to manage knowledge. One of the main practices when it comes to KM is to create an "Experience Factory". In a software development environment the experience from each activitiy conducted can be collected and packaged and stored in an experience base to be easily reused, documented, and thus it can be accessed by multiple organization members. Explicit knowledge can be described by means of drawings and writing, consequently it can be easily used and shared [41].

As of 2000, one can say that the number of actions that produced research in KM and Software Engineering increased considerably. In 2002, a special issue of IEEE Software was devoted to KM in Software Engineering [33]. In 2003, the book "Managing Software Engineering Knowledge" [3] was published, focusing on a range of topics on the subject. PhD theses have also been published over the years, for instance, the studies presented in [11, 51]. In addition to the published primary studies presenting the conduction of empirical studies on the subject, it is also possible to find several secondary studies addressing broad overviews of KM in software engineering [1, 4, 52].

2.2 Knowledge Management Diagnostic

There are KM cycles that propose an activities set in order to promote KM initiatives [9]. Some main KM cycle activities are capturing, creating, encoding, sharing, accessing, applying and reusing the knowledge. However, while there are several KM approaches, some authors emphasize that conducting an organization diagnostic can identifying which organization area or knowledge activities really need improvement or presents with most cost-effective for organization in terms of KM [5, 48].

A knowledge diagnostic will help the audited organization to determine which knowledge is being managed and how well it is being managed. Audit helps to make the knowledge in the company visible. KM diagnostic has been introduced as an effective process to monitor the performance of KM practices in an organization [37]. Usually the diagnosis is conducted through some type of research instrument, most of the time questionnaires or interviews are used [14]. The issues are measured and the results show where the organization can guide its KM efforts.

2.3 Related Works

In order to find more reviews that presented KM diagnostics we conducted a tertiary study, however searching for other studies considered secondary (Systematic Literature Review or Systematic Mapping). Tertiary studies are considered as a review that focuses only on secondary studies [24]. In this tertiary study, we used the following search string: *("Knowledge Management") AND ("diagnostic" OR "diagnose" OR "diagnosis" OR "audit" OR "assessment" OR "evaluation") AND ("systematic literature review" OR "systematic review" OR "systematic mapping" OR "mapping study" OR "systematic literature mapping" OR "literature review").*

The search string was applied in the following electronic databases: *IEEE Xplore, ACM Digital Library, Scopus* and *Science Direct*. A total of 262 studies were returned. Nevertheless, we did not find any secondary study that addressed KM diagnostic. Thus, we conducted a manual search in order to identify some reviews that addressed the same scope of this research and we found two studies that attracted our attention.

In Pa et al. [44], frameworks, models, methodologies, processes and techniques for Knowledge Audit (KA) were investigated. From the study results it is clear that different approaches have been proposed for KA, mainly methodologies and techniques. Even so, there is no consensus on the approaches involved in KA implementation. The main differences of the review conducted by Pa et al. and ours are that we used a string that covers more synonymous for diagnosis (Table 1). Another major difference is that our scope is only the software development organizations. A more general scope was considered in [44].

In the study conducted by Esteves [14], a tradicional review of characteristics of models most used for KM diagnostic in the Brazilian context was conducted. A method (e.g., questionnaires, interviews) used in a KM diagnostic usually follows some diagnostic model already known in the literature. According to Esteves [14], in Brazilian organizations the most commonly models used for KM diagnostic are *Asian Productivity Organization* (APO)[1], *Organizational Knowledge Assessment* (OKA) [15], Seven Dimensions of Knowledge Management [55] and Knowledge Management Diagnostic (KMD) [5]. In [14] also are presented some tools for KM diagnostic, such as Knowledge Management Assessment Tool (KMAT), SysOKA (based on OKA model), American Productivity and Quality Center (APQC). However, such models and tools mentioned in [14] have been applied mainly in federal public administration environments [14]. No mention is made about software development organizations.

3 RESEARCH METHOD

SLR has been used to provide a complete evaluation of the state of all relevant research available for a specific topic of interest [24]. This review involves three main phases: (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. In this phase the SLR findings are used to answer the research questions.

In addition to the searches in the databases, we also used backward snowballing [24] in order to identify additional relevant studies through the reference lists of the studies found using the search strings. We used this approache in our mapping to overcome the limitation of using a specific set of electronic databases.

Following a summary of protocol[2] is presented conducted in this SLR.

Research Questions. This SLR was conducted in order to identify KM diagnostic in software development organizations. Based on this goal, two Research Questions (RQs) were created:

RQ1. What is the purpose of applying a KM diagnostic in software engineering context?

RQ2. What are the main diagnostic characteristics?

Search String. In order to identify the terms for the search string, a manual review was initially performed. In the manual review the proceedings of International Conference on Software Engineering and Knowledge Engineering (SEKE)[3]

[1] http://www.apo-tokyo.org/
[2] The complete protocol is available in *https://goo.gl/hM2cY7*
[3] *http://www.ksi.edu/seke/skhistory.html*

were analyzed between the editions from 2004 to 2017, looking for studies that presented KM diagnostic approaches in Software Engineering. SEKE is one of the leading conferences that present relevant research results in Software Engineering and Knowledge Engineering. This event has been one of the main arenas for empirical studies as well as technological development related to knowledge management in software engineering [4].

In the manual review, we found one study that presented some kind of KM analysis activity in the software organization. In [56], the deployment architecture process was analyzed from a KM perspective, using a KM audit methodology. The analysis aims was identified KM gaps, which can cause deployment requirements' traceability (RT) problems.

In addition to study identified in SEKE, we also had two more studies as a control group to create and validate the search string created. If the publications of the control group are not retrieved, the string needs to be calibrated (adjusted) or new databases need to be added in order to guarantee that at least these publications are retrieved. In this sense, we used a control group to calibrate the search string.

The search string considers three areas – “Knowledge Management”, “Diagnosis” and “Software Engineering” - and it was applied in three metadata fields: title, abstract and keywords. Table 1 presents the areas, keywords and final search string created.

Sources. The databases used in this research were: IEEE Xplore, ACM Digital Library, ScienceDirect and Scopus.

Selection Criteria. The selection criteria are organized in one Inclusion Criterion (IC) and five Exclusion Criteria (EC). The inclusion criterion is: **(IC1)** The study must present existing research on KM diagnostic in software development organizations. 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 primary study, such as editorials, summaries of keynotes, workshops, and tutorials; and **(EC5)** The full paper is not available.

4 SELECTION PROCESS

The selection process was divided into four main stages, as shown in Figure 1. Figure 1 shows how many articles remained in each selection process stage.

Using the search string, 1012 were retrieved. The selection process applied on the returned publications was performed in four stages. In the 1^{st} stage duplicated studies were eliminated, resulting in 926 publications (reducing approximately 8.5%). In the 2^{nd} stage, the selection criteria (inclusion and exclusion criteria) were applied for title, abstract and keywords, leading to 224 studies (reducing approximately 75.8%). In the 3^{rd} stage, the selection criteria were applied considering the full text, resulting in a set of 18 studies (reducing approximately 92%). Over these 18 studies considered relevant, we performed backward snowballing in 4^{th} stage, which resulted in 38 studies. From these 38 studies, the selection criteria were applied considering the title, abstract and keywords. Next, the selection criteria were applied considering the full text. A total of 5 studies were resulted from this stage. As a result, we got to 23 studies (18 from the sources and 5 from snowballing) from selection process. conAdding the article that returned from the manual review in proceedings of SEKE conference, the final result was 24 studies. The complete references of the 24 selected studies can be accessed in *https://goo.gl/WJcyG4*.

5 DATA EXTRACTION AND SYNTHESIS

After selecting the primary studies, we analyzed each one in order to answer the research questions, with attention to the following items: study purpose to diagnose the KM in a software development organization and format used for the diagnostic, such as application, investigated items, tool used and the model on which diagnostic was based. Next, we present the data extraction and synthesis regarding these research questions.

RQ1. *What is the purpose of applying a KM diagnostic in software engineering context?*

In the first research question, our intention was to understand the studies purpose in apply a KM diagnostic and in what software engineering context. Some studies have a very specific purpose for applying a KM diagnostic, for example, improve the Requirement Traceability (RT) in the organization or improve organizational culture. However, the majority of the studies conducted have a more general purpose in the organization, for instance, Software Process Improvement (SPI). Some of these purposes are described below.

Software Process Improvement (SPI) is discussed in [2, 8, 8, 19–21, 35, 36, 45, 47]. SPI is a continuous modification of a software development process for the purposes of reducing production costs, improving schedule adherence and increasing product quality. According to Mitchell and Seaman [36], KM can be viewed as complementary to SPI when used in a software development domain. Indeed, capturing past software project knowledge for use in subsequent projects is a common SPI technique, especially, when there is an effective and efficient use and flow of knowledge, during software product development [35]. The utilization of an approach for assessing Organizational Learning (OL) also provides a basis for SPI [8]. OL is considered the process of learning by individuals and groups in a software development organization through the software development process or KM [8].

Some studies had as specific purpose the organizational culture improvement [28, 29, 47]. Organizational culture is a means to ensure that stakeholders will be in tune and working towards achieving the same objectives. Organizational culture directly impacts the success of KM, since it influences the way employees learn and share knowledge in the organization [47]. According to Levy et al. [28], an organization's culture must be understood before KM model can be successfully implemented. For this a diagnosis of KM can be conducted.

Table 1: Areas and Keywords

Areas	Keywords
Knowledge Management	"Knowledge Management"
Diagnosis	"diagnostic", "diagnose", "diagnosis", "audit", "assessment", "evaluation")
Software Engineering	"Software Engineering", "software development", "software engineering", "software process", "development companies", "software product", "software quality"
Search String: ("Knowledge Management") AND ("diagnostic" OR "diagnose" OR "diagnosis" OR "audit" OR "assessment" OR "evaluation") AND ("Software Engineering" OR "software development" OR "software engineering" OR "software process" OR "development companies" OR "software product" OR "software quality")	

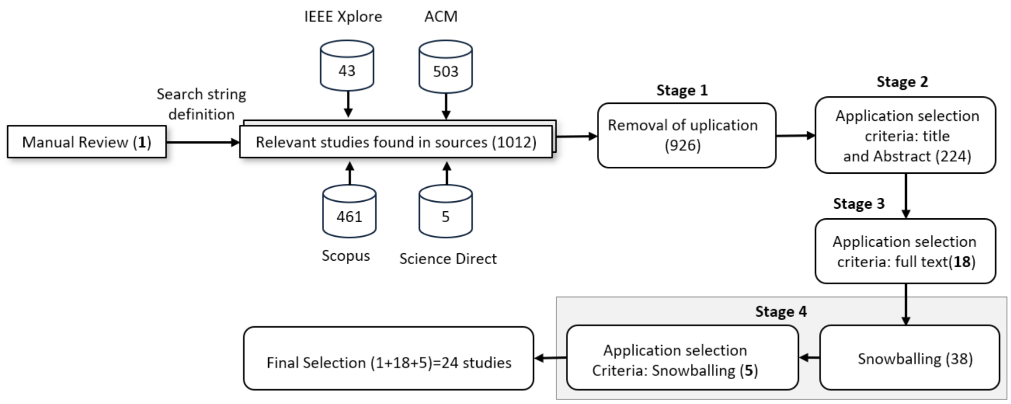

Figure 1: Search and selection review process.

To be successful, the KM approach must support cultural organization practices.

Another purpose for performing a KM diagnostic discussed is the Requirement Traceability (RT) improvement [56]. According to Shpigel and Hadar [56], an important aspect of the software development process is RT. RT is recognized as a concern in guidelines and standards in requirement engineering. However, some organizational factors can undermine this process, for example, deployment architecture process. A KM diagnostic on these factors may help to identify several KM gaps and thus avoid deployment RT problems. Still referring to the software requirements, the purpose of the studies [53] and [54], focuses on knowledge audit in their requirement elicitation process. Requirement elicitation process involves a great deal of knowledge and there are several problems regarding eliciting and using the knowledge in this process. Thus, in order to improve the requirement elicitation, it is important to identify knowledge components and knowledge sources existing in the requirement elicitation process from an audit of knowledge.

In [12, 16, 25] the purpose is related to contributing with the agile practices in software development in terms of communication between teams, knowledge sharing, documentation and risk-strategy analysis. Knowledge sharing, for example, is difficult for distributed agile teams due to spatial, temporal, and cultural barriers, which negatively affect face-to-face interaction, communication and collaboration. Thus, understanding and assessing on how do agile teams gather, store, share, and use knowledge in distributed software development can contribute to successful implementation and quality of distributed agile projects.

In addition to the purposes mentioned above and described in more detail, other purposes have also been identified, such as: software quality enhancement by customer KM in software companies [23]; the teams' knowledge transfer effectiveness [18]; analysis of integration of KM techniques into the activity of risk management [16, 39]; and Assessment of KM's structure [45].

RQ2. *What are the main diagnostic characteristics?*

This research question help us to understand how the studies conducted the analysis in the organization with respect to KM and especially what approaches already known in the literature were used or what new approhes were proposed. Table 2 presents a summary of the approaches used in the selected studies to conduct KM diagnostic.

Most of the studies analyzed showed that the approaches used for the diagnostic were not based on some model or process already existing in the literature. The conducted KM diagnostic was defined by the authors, themselves, of the returned studies. For example, in [47] one of the objectives was to investigate KM practices in use by the software organization. In order to identify practices, the authors conducted interviews. The interview focused on investigating how KM and OL take place in the organization. In [2] a questionnaire and interviews were developed to identify current practice of KM in Software Engineering processes in two Australian companies on the basis that they both claimed to apply KM

Table 2: Approaches used in the selected studies to conduct KM diagnostic.

Study	Year	Knowledge items investigated	Application Method	Model	Tool
[2]	2008	KM practices	Interviews	None	Not mentioned
[8]	2013	Organizational Learning (OL)	Questionnaire, Interviews	AiOLoS (Assessing OL of software development organizations)	Not mentioned
[12]	2012	KM in Distributed Software Development	Interviews	None	Not mentioned
[16]	2018	Knowledge sharing, Risk Management Model	Interview	None	Not mentioned
[17]	2010	Knowledge management metrics	Interview	None	Not mentioned
[18]	2018	Teams' knowledge transfer, Global Software Development (GSD)	Questionnaire	None	Not mentioned
[19]	2004	Knowledge flow (K-flow)	Audio-recorded, Interviews	"Knowledge map" or K-map	Not mentioned
[20]	2009	Knowledge sources, tools, techniques, methods applied in the organization, KM activities, tacit knowledge and rate the level of effort invested on KM	Interviews	None	Not mentioned
[21]	2012	Experience capture	Questionnaire	None	Not mentioned
[23]	2018	Customer Knowledge Management (CKM)	Questionnaire	None	Not mentioned
[25]	2013	Organizational Learning (OL)	Questionnaire, Interviews	AiOLoS (Assessing OL of software development organizations)	SurveyMonkey
[26]	2009	KM sucess and KM service	Questionnaire	None	Not mentioned
[28]	2008	Tacit cultural perceptions	Questionnaire, Interviews	None	Not mentioned
[29]	2010	Tacit cultural perceptions	Questionnaire, Interviews	CommonKADS	Not mentioned
[35]	2011	Knowledge flow (K-flow)	Interviews, Artifacts	"Knowledge map" or K-map	Not mentioned
[36]	2016	Knowledge flow (K-flow)	Questionnaire	None	Not mentioned
[39]	2014	Risk Management, KM techniques	Questionnaire, Observation, Interviews, Documents	None	Not mentioned
[43]	2013	Organizational Learning (OL)	Questionnaire, Interviews	AiOLoS (Assessing OL of software development organizations)	Not mentioned
[45]	2002	Knowledge processes	Maturity model stages	Knowledge Process Quality Model (KPQM)	Not mentioned
[47]	2015	Organization Cultural	Interviews, Observations of the LL meetings	Organizational Culture Assessment Instrument (OCAI)	Not mentioned
[50]	2008	Organizational Leadership	Questionnaire	None	Knowledge Management Assessment Tool (KMAT)
[53]	2014	Requirement Elicitation, Knowledge components, knowledge sources	Questionnaire	Knowledge audit model (proposed by the study)	Not mentioned
[54]	2015	Requirement elicitation process, Knowledge communication	Questionnaire	Knowledge audit model (proposed by the study)	Not mentioned
[56]	2013	Deployment architecture	Questionnaire, Interviews	Methodology SEKAM	Not mentioned

practices in their software development work. [28] conducted structured interviews in order to capture tacit cultural perceptions of employees in order to identify barriers that might affect the adoption of KM solutions. However, none of these authors mentioned how interview or questionnaires questions were created.

In summary, almost all the selected studies were based on interviews and questionnaires to analyze KM applied in the organization, totaling 62.5% of studies each one and 29.2% applied two approaches (questionnaire and interviews) together (Figure 2). Other approaches applied were audio-recorded [19], artifact analysis [35, 39], observations of the LL meetings [39, 47] and based on maturity model stages [45]. In some studies, the authors mention that the questions elaborated in the questionnaires or interviews were defined based on bibliographic reviews to identify the factors, techniques or practices used in software engineering and KM [18, 23, 25, 39, 53, 54]. Other authors mention having based on Goal/Question/Metric (GQM) approach [8, 25, 43] or International Organization for Standardization (ISO) [45]. Although GQM or international standard are not a specific model to apply KM diagnostic, these approach are strong references to help in the elaboration of questions for questionnaires or interviews.

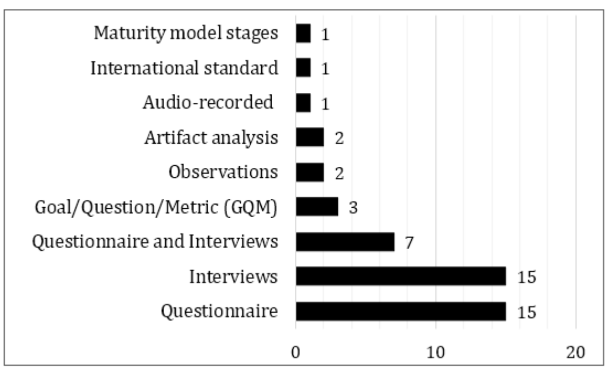

Figure 2: Application Method

From the analysis of the 24 selected studies, 11 use some approach, such as model or process, in order to conduct the diagnostic in the organization. Out of these 11 studies, seven different approaches were identified: Knowledge audit model proposed in [53, 54]; AiOLoS [8, 25, 43]; CommonKADS [29]; K-map [19, 35]; Organizational Culture Assessment Instrument (OCAI) [47]; Knowledge Process Quality Model (KPQM) [45]; and Methodology SEKAM [56]. The identified approaches are presented briefly below.

A knowledge audit model was proposed in [53, 54]. The model aims to improve the requirement elicitation process by identifying knowledge components and knowledge sources existing in the requirement elicitation process as well as their relationships. The model proposed is based on Iterative Triangulation Method [31]. Basically, an iterative triangulation makes use of iteration between literature review, case studies and intuition to develop a new theory. This method consists of four phases: (i) Groundwork: Literature review and case selection; (ii) Induction: Cases analysis and shaping conjectures; (iii) Iteration: Theory refinement; (iv) Conclusion: Theory Evaluation and suggesting future research.

In [8, 25] and [43] a model for assessing the level and characteristics of OL in software development organizations (AiOLoS) is proposed. AiOLoS provides a framework for comparison among software organizations with respect to their OL capabilities, to allow these organizations to identify their deficiencies, offering the means for the measurement of the realized improvement in OL and provide a starting point for SPI. AiOLoS model consists of three major process areas that map to the three major objectives of a learning software organization: obtaining, using and passing knowledge. In order to assess these areas in a software organization within the proposed model, GQM approach was applied.

[29] incorporated a knowledge engineering methodology, called CommonKADS, for auditing purposes. CommonKADS is the methodology to support structured knowledge engineering [49]. This methodology offers a set of steps to support the specification and development of knowledge-based systems. In addition, the methodology can also be designed to measure KM in an organization since offers instruments and methods that aim to generate more organization knowledge sharing, as well as reducing the time in the development of new projects, due to its great reusability. The methodology also covers several design aspects of a knowledge-based systems, including: organizational analysis, project management; acquisition, representation and modeling of knowledge; systems integration and implementation.

Knowledge mapping (or K-maps) has been used to characterize the knowledge within an organization with documents or illustrations depending on the intended use. For example, there are online employee "yellow pages" and organizational charts to help employees locate others who may be able to answer their questions. In [35] and [19], K-maps are used to for mapping the organizational knowledge flows in a software organization by utilizing the people perspectives and artifact analysis. K-maps can be used as diagnostic since they are used to understand the current situation of the project (or organization) and thereby provide an overview of the current status of the knowledge flows in organizations mapped situation enabling a closer examination.

An investigation on cultural profile of the software organization was conducted in [47]. The approach used was Organizational Culture Assessment Instrument (OCAI) [6]. This instrument is a questionnaire that requires individuals to respond just six items: dominant characteristics; organizational leadership; management of employees; organization glue; strategic emphases; and, criteria of success. Each item has four alternatives related to the organization cultures.

[45] proposed a new model called Knowledge Process Quality Model (KPQM). This model is based on the ideas of quality management and process engineering and helps organizations to assess and improve their KM structures to control knowledge processes. KPQM was also constructed based on SPICE methodology [13] from an adaptation of the

six stages of maturity that the methodology proposes: 0 - incomplete, 1 - performed, 2 - managed, 3 - established, 4 - predictable and 5 - optimizing.

Finally, an audit methodology, called Socio-Engineering Knowledge Audit Methodology (SEKAM), is used in [56] in order to analyze the deployment architecture process. SEKAM methodology identifies KM requirements within a knowledge intensive business process [27]. SEKAM is based on five stages: Organizational Analysis; Define Audit Project Properties; Knowledge Inventory of the Business Process; Knowledge Inventory of the Business Process; and Audit Results Approval. Each of the SEKAM steps is based on practical knowledge modeling instruments for information elicitation and analysis. The data collection to be applied and analyzed in the steps proposed by the methodology is based on questionnaires and interviews.

Although strongly present in other areas, such as administration, well-known models for KM diagnostic, such as those mentioned in Section 2.2, in the context of software engineering this practice does not yet appear to be well consolidated. Over 54% of the studies selected conducted questionnaires and interviews without using existing and validated approaches in the literature to create the questions. We also note that some studies that were returned in our SLR propose new approaches, such as those presented in [8, 25, 43, 53, 54]. So, only 33% of studies were based on already existing approaches in the literature to conduct a diagnostic in a software organization in terms of KM. It is believed that, this scenario is due to the fact that this type of research in software engineering is relatively new. Out of the 24 studies selected in this SLR, the oldest study is from 2002, as can be seen in Table 2.

Regarding the tools used, only two papers mention their use to automate the process of applying the diagnostic. Other studies did not use or at least did not mention their use. In [50], Knowledge Management Assessment Tool (KMAT) is used. KMAT is a collaborative benchmarking tool, designed to help organizations make an initial high-level assessment about KM [22]. KMAT intention is collaborative benchmarking. KMAT presents a questionnaire composed of five sections of evaluation: leadership (to verify if KM is compatible with how the organization is managed); technology (how the organization facilitates communication among individuals in collecting, storing, and sharing knowledge); culture (the culture of knowledge sharing is explicit among individuals); measurement (how the organization quantifies its knowledge capital); process (concentrates on activities to create, identify, collect, adapt, organize, apply and share knowledge). On the other hand, although it is not a specific tool for KM diagnostic, in [25] the SurveyMonkey tool is used to collect KM data within the organization. SurveyMonkey[4] is a worldwide platform of questionnaires that facilitate the gathering of people's opinions and can be turned into insights.

Completing our analysis on approaches used for KM diagnostic in the selected studies, we analyzed the main knowledge items investigated by the studies. Many are related to answers of RQ1. As can be observed in Table 2, several different knowledge items were investigated, for example, Customer Knowledge Management (CKM), Requirement elicitation, Risk Management, Organizational Learning (OL), Organizational Cultural and Knowledge flow (K-flow); the last two being those most investigated. In relation to how these items have been investigated in the studies considering the knowledge type (tacit or explicit), most studies deal with the two types. Figure 3 show this proportion. Although, the literature points out that the most valuable knowledge within an organization is essentially tacit [12], in this SLR the studies showed that there is a great concern with explicit knowledge.

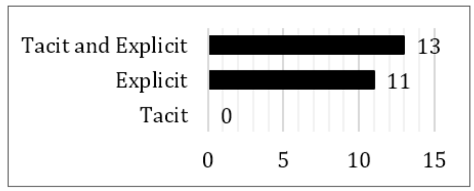

Figure 3: Knowledge types analyzed

6 DISCUSSION

The objective of this work was to identify and summarize the main approaches used to diagnose KM in software development organizations. We found studies that are directly related to diagnostic or audits in relation to KM, but we also identified studies that implicitly lead to such approaches. Such studies have aided us to infer about how software development organizations have conducted KM diagnostic.

Although KM brings many benefits, there are also problems: KM systems are not appropriate yet; employees are normally reluctant to share their knowledge; and increased workload [52]. Therefore, organizations, such as those from software development, have a limited budget to invest in KM. KM diagnostic, when performed, is conducted only in specific items which hold the greatest potential for future growth and strategic advantage [5]. As identified in the studies selected in this SLR, specific KM items are investigated (Table 2). Thus, the organization can invest in items where KM can have more potential and thus avoid high and often ineffective investments. The diagnostic of knowledge can minimize the effort and time spent in incorporating KM approaches. Ideally, auditing should be an ongoing process.

For modern organizations knowledge is a key parameter for surviving, since it enables a continuous improvement of services, organizational culture and software product. This is due to the fact that this parameter relies on the abilities to incorporate the earlier experiences into the planning of future practice. These abilities depend highly on the organization's ability to create and make the knowledge flows. One of the main items investigated by the selected studies was K-flow. K-flow provides insight to where ideas or initiatives originated and how knowledge about these spread to the rest of the organization [19]. KM is a systematic approach to help K-flow in the organization.

[4] *https://pt.surveymonkey.com/*

The studies selected in this SLR also presented different approaches for KM diagnostic that have been used for different purposes in software organizations. However, as already highlighted in RQ2, this practice does not appear to be well consolidated. Many studies conducted an analysis in the organization without using already existing and validated approaches. The main instrument for conducting the diagnostic has been based on interviews and questionnaires with the organization members. In order to determine how the questions should be constructed, the literature is used as a basis by some authors. In addition, only two studies mention automation of the diagnostic process in terms of application and compilation of collected data and the knowledge type more investigated by the selected studies is the knowledge explicit.

KM diagnostic can reveal an organization's knowledge strengths, weaknesses, opportunities, threats and risks [32]. Software organizations can use KM in order to guarantee and reuse the knowledge. Thus, the main contribution of this SLR was to provide the understanding and making evident aspects associated to the KM diagnostic in software organizations. Answers to RQs can help practitioners learn about successes used in the context of organizations that employ same approaches. The research results will also help area researchers identifying future research as well as provide a direction to appropriately position new research activities in KM diagnostic and software engineering.

6.1 Limitations

This SLR has some limitations. The study selection and data extraction stages were initially performed by the first paper author, and thus some subjectivity could have been embedded. In order to reduce this subjectivity, the other authors performed these same steps over different sample of studies. The results of each reviewer were then compared in order to detect possible bias.

Our review was limited by the search terms used and the electronic databases included. We tried to overcome the limitations by using a manual review on SEKE proceedings in order to identify the string terms. We also consider articles from a control group to calibrate the string.

Regarding the factor related to how the research area is defined, this factor also posed difficulties in defining the search string. Initially, we considered only studies that mention the terms "diagnostic", "diagnose" and "diagnosis". However, it is not enough for characterizing the studies in the area. Thus, in our research string we also consider terms as "audit", "assessment" or "evaluation" trying to broadly cover the area. Even so, because we considered studies indexed just by the selected electronic databases, and those obtained from snowballing, hence, possibly leaving out some valuable studies for our analysis. However, the studies discussed in this SLR provide a snapshot of empirical research on outcomes and impacts of existing research on KM diagnostic and software engineering.

7 CONCLUSIONS

In this paper, we presented a SLR in order to summarize existing research on KM diagnostic in software organizations. We identified 24 studies addressing investigated KM diagnostic. From these 24 studies selected, seven different KM diagnostic were identified. The other studies conducted questionnaires or interviews without a consolidated reference.

From this SLR, we highlight the following conclusions: (i) KM diagnostic is an effective process to monitor the performance of KM practices; (ii) in the software engineering context the KM diagnostic practice does not yet appear to be well consolidated; (iii) questionnaires and interviews are very used for KM diagnostic; (iv) few studies, only two, use tools to automate the process of applying the diagnostic; and (v) there is a great concern with explicit knowledge.

As a future work, we intend to continue to study existing approaches to KM diagnostic, especially in areas that have this theme more consolidated, and we also intend to bring these approaches to the reality of software development organizations.

REFERENCES

[1] Y. Andriyani, R. Hoda, and R. Amor. 2017. Understanding Knowledge Management in Agile Software Development Practice. International Conference on Knowledge Science, Engineering and Management (KSEM), 195–207.

[2] A. Aurum, F. Daneshgar, and J. Ward. 2008. Investigating Knowledge Management practices in software development organisations - An Australian experience. *Information and Software Technology* 50 (2008), 511–533.

[3] Ay. Aurum, J. Ross, W. Claes, and M. Handzic. 2003. *Managing Software Engineering Knowledge*. Springer-Verlag, Berlin, Heidelberg.

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

[5] W. Bukowitz and R. L. Williams. 2000. *The knowledge management fieldbook*. Financial Times Prentice Hall, Great Britain.

[6] K. S. Cameron and R. E. Quinn. 2006. *Diagnosing and Changing Organisational Culture: The Competing Values Framework*. John Wiley & Sons.

[7] J. B. Carreteiro, P. and. Vasconcelos, A. Barão, and A. Rocha. 2016. A Knowledge Management Approach for Software Engineering Projects Development. In *New Advances in Information Systems and Technologies*. Springer International Publishing, 59–68.

[8] O. Chouseinoglou, D. Iren, N. A. Karagoz, and S. Bilgen. 2013. AiOLoS: A model for assessing organizational learning in software development organizations. *Information and Software Technology* 55 (2013), 1904–1924.

[9] K. Dalkir. 2005. *Knowledge Management in Theory and Practice*. Elsevier, Burlington, MA.

[10] T. H. Davenport and L. Prusak. 2000. *Working knowledge* (2 ed.). Harward Business School Press" Boston, USA.

[11] F. O. Dingsøyr. 2007. *Knowledge Management in Software Process Improvement*. Ph.D. Dissertation. Norwegian University of Science and Technology, Department of Computer and Information Science Faculty of Information Technology, Mathematics and Electrical Engineering.

[12] S. Dorairaj, J. Noble, and P. Malik. 2012. Knowledge Management in Distributed Agile Software Development. Agile Conference, 64–73.

[13] K. El Emam, J.N. Drouin, W. Melo, and A. Dorling. 1998. *SPICE - The Theory and Practice of Software Process Improvement and Capability Determination*. Wiley-IEEE Computer Society Press, Los Alamitos (CA).

[14] S. R. M. Esteves. 2017. *Requisitos de software funcionais para o desenvolvimento de plataforma digital de diagóstigo da gestão*

do conhecimento nas organizações. Masters Dissertation (In Portuguese). UniCesumar, Maringá, Paraná, Brazil.

[15] A. F. Fonseca. 2006. *Organizational Knowledge Assessment Methodology* (2 ed.). World Bank Institute, Washington.

[16] S. Ghobadi and L. Mathiassen. 2018. Risks to Effective Knowledge Sharing in Agile Software Teams: A Model for Assessing and Mitigating Risks. *Information Systems Journal* 27 (2018), 699–731.

[17] V. Goldoni and M. Oliveira. 2010. Knowledge management metrics in software development companies in Brazil. *Journal of Knowledge Management* 14 (2010), 301–313.

[18] J. Gopal, A. K. Sangaiah, A. Basu, and X. Z. Gao. 2018. Integration of fuzzy DEMATEL and FMCDM approach for evaluating knowledge transfer effectiveness with reference to GSD project outcome. *International Journal of Machine Learning and Cybernetics* 9 (2018), 225–241.

[19] B. H. Hansen and K. Kautz. 2004. Knowledge Mapping: A Technique for Identifying Knowledge Flows in Software Organisations. European Conference on Software Process Improvement, 126–137.

[20] S. Iuliana. 2009. A knowledge management practice investigation in Romanian software development organizations. *WSEAS Transactions on Computers* 8 (2009), 459–468.

[21] M. Ivarsson and T. Gorscherk. 2012. Praction Selection Framework. *International Journal of Software Engineering and Knowledge Engineering* 22 (2012), 17–58.

[22] M. Jager. 1999. The KMAT: benchmarking knowledge management. *Library Management* 20 (1999), 367–372.

[23] A. Khosravi, A. R. C. Hussin, and M. Nilashi. 2018. Toward software quality enhancement by Customer Knowledge Management in software companies. *Telematics and Informatics* 35 (2018), 18–37.

[24] B. A. Kitchenham and S. Charters. 2007. *Guidelines for performing Systematic Literature Reviews in Software Engineering*. Technical Report EBSE 2007-001. Keele University and Durham University, UK.

[25] L. Lagerberg, P. Emanuelsson, K. Sandahl, and D. Stahl. 2013. The impact of agile principles and practices on largescale software development projects: A multiple-case study of two projects at Ericsson. International Symposium on Empirical Software Engineering and Measurement (ESEM), 348–356.

[26] F. Lehner. 2009. Measuring KM Success and KM Service Quality with KnowMetrix – First Experiences from a Case Study in a Software Company. 3rd International Conference on Knowledge Science, Engineering and Management, 335 – 346.

[27] M. Levy, I. Hadar, and I. Aviv. 2009. Enhancing Knowledge Intensive Business Processes via Knowledge Management Audit. Americas Conference on Information Systems, California.

[28] M. Levy, I. Hardar, S. Greenspan, and E. Hadar. 2008. Knowledge Management Culture Audit: Capturing Tacit Perceptions and Barriers. Americas' Conference on Information Systems (AMCIS), 1–8.

[29] M. Levy, I. Hardar, S. Greenspan, and E. Hadar. 2010. Uncovering cultural perceptions and barriers during knowledge audit. *Journal Knowledge Management* 14 (2010), 114–127.

[30] M. Levy and O. Hazzan. 2009. Knowledge management in practice: The case of agile software development. In *Cooperative and Human Aspects on Software Engineering*. IEEE, 60–65.

[31] M. Lewis. 1998. Iterative triangulation: a theory development process using existing case studies. *J. Oper. Manag.* 16 (1998), 455–469.

[32] J. Liebowitz. 2012. *Knowledge Management Handbook:Collaboration and Social Networking,* (2 ed.). RC Press.

[33] M. Lindvall and I. Rus. 2002. Knowledge management in software engineering. *IEEE Software* 19 (2002), 26–38.

[34] C. P. C. Maciel, E. F. Souza, N. L. Vijaykumar, R. A. Falbo, G. V. Meinerz, and K. R. Felizardo. 2018. An Empirical Study on the Knowledge Management Practice in Software Testing. In *Experimental Software Engineering Latin American Workshop (ESELAW'18). XXI Ibero-American Conference on Software Engineering (CIBSE)*.

[35] S. M. Mitchell and C. B. Seaman. 2011. A Knowledge Mapping Technique for Project-level Knowledge Flow Analysis. International Symposium on Empirical Software Engineering and Measurement, 347–350.

[36] S. M. Mitchell and C. B. Seaman. 2016. Could removal of project-level knowledge flow obstacles contribute to software process improvement? A study of software engineer perceptions. *Information and Software Technology* 72 (2016), 151–170.

[37] M. Nejati. 2010. Knowledge Management Performance Evaluation: Challenges and Requirements for Organizations. *Technics Technologies Education Management* 5 (2010), 251–254.

[38] S. Nerur and V. Balijepally. 2007. Theoretical reflections on agile development methodologies. *Commun. ACM* 50 (2007), 79–83.

[39] S. M. Neves, C. E. S. Silva, V. A. P. Salomon, A. F. Silva, and B. E. P. Sotomonte. 2014. Risk management in software projects through Knowledge Management techniques: Cases in Brazilian Incubated Technology-Based Firms. *International Journal of Project Management* 32 (2014), 125–138.

[40] I. Nonaka and G. Krogh. 2009. Tacit Knowledge and Knowledge Conversion: controversy and Advancement in Organizational Knowledge Creation Theory. *Organization Science* 30 (2009), 635–652.

[41] I. Nonaka and H. Takeuchi. 1997. *The knowledge-creating company*. Oxford University Press, Oxford, New York.

[42] D.E. O'Leary and R. Studer. 2001. Knowledge Management: an Interdisciplinary Approach. *IEEE Intelligent Systems* 16, No. 1 (2001).

[43] G. Ozen, N. A. Karagoz, O. Chouseinoglou, and S. Bilgen. 2013. Assessing Organizational Learning in IT Organizations: An experience report from industry. Intern. Workshop on Software Measurement (IWSM) and International Conference on Software Process and Product Measurement (Mensura), 253–258.

[44] N. C. Pa, A. Taheri, and R. Abdullah. 2012. A Survey on Approaches in Knowledge Audit in Organizations. *Asian Transactions on Computers* 02 (2012).

[45] O. Paulzen, M. Doumi, P. Perc, and A. Roibas. 2002. A Maturity Model for Quality Improvement in Knowledge Management. International Symposium on Empirical Software Engineering and MeasurementEnabling Organizations and Society through Information Systems (ACIS), 243–253.

[46] J. Pfeiffer and R. Sutton. 1999. *The knowledge-creating company*. Harvard Business School Press, Boston.

[47] J. H. Rabelo, E. C.C. Oliveira, D. V. Santos, L. C. S. Braga, G. S. Souza, I. F. Steinmacher, and T. U. Conte. 2015. Knowledge Management and Organizational Culture in a Software Organization – a Case Study. 8th International Workshop on Cooperative and Human Aspects of Software Engineering, 89–92.

[48] O. M. Rodriguez-Elias, A. I. Martínez-García, A. Vizcaíno, J. Favela, and M. Piattini. 2008. A framework to analyze information systems as knowledge flow facilitators. *Information and Software Technology* 50 (2008), 481–498.

[49] G. Schreiber, H. Akkermans, A. Anjewierden, R. De Hoog, N. Shadbolt, W.V. De Velde, and B. Wielinga. 1999. *Knowledge Engineering and Management: The CommonKADS Methodology*. MIT Press, Cambridge, MA.

[50] S. K. Singh. 2008. Role of leadership in knowledge management: a study. *Journal of Knowledge Management* 12 (2008), 3–15.

[51] E. F. Souza. 2007. *Knowledge management applied to software testing: an ontology based framework*. Ph.D. Dissertation. National Institute for Space Research (INPE), Applied Computing, São José dos Campos, Brazil.

[52] E. F. Souza, R. A. Falbo, and N. L. Vijaykumar. 2015. Knowledge management initiatives in software testing: A mapping study. *Information and Software Technology* 57 (2015), 378–391.

[53] L. Taheri, N. C. Pa. Abdullah, S. Abdullah, and Shafazand M. Y. 2014. Identifying Knowledge Components in Software Requirement Elicitation. International Conference on Industrial Engineering and Engineering Management, 286–291.

[54] L. Taheri, N. C. Pa, R. Abdullah, and S Abdullah. 2015. A Knowledge Audit Model to Assess the Knowledge in Requirement Elicitation Process. 9th Malaysian Software Engineering Conference, 106–111.

[55] J. C. C. Terra. 2001. *Gestão do Conhecimento: O Grande Desafio Empresarial* (2 ed.). Negócio Editora, São Paulo.

[56] N. Unkelos-Shpigel and I. Hadar. 2013. Enhancing Deployment Requirements' Traceability via Knowledge Management Audit. In *The 25th International Conference on Software Engineering & Knowledge Engineering (SEKE)*. 574–577.

[57] S. Vasanthapriyan, J. Tian, and J. Xiang. 2015. A Survey on Knowledge Management in Software Engineering. International Conference on Software Quality, Reliability and Security Companion (QRS-C), Vancouver, BC, Canada, 237–244.