



+62 813 8765 4578

+62 813 8765 4578

<https://dinastipub.org/DIJDBM>

dinasti-info@gmail.com

THE ANALYSIS OF KNOWLEDGE MANAGEMENT PROCESS ON SOFTWARE DEVELOPMENT PROCESS: A SYSTEMATIC REVIEW

Arief Ramadhan¹, Lila Setiyani²

¹⁾ STMIK ROSMA, West Java, Indonesia

²⁾ STMIK ROSMA, West Java, Indonesia

ARTICLE INFORMATION

Received: 16th March 2020

Revised: 20th April 2020

Issued: 30th June 2020

Corresponding author:
first author

E-mail:

lila.setiyani@dosen.rosma.ac.id

arief.ramadhan@rosma.ac.id



DOI:10.31933/DIJDBM

Abstract: Software quality is affected by its development process. The software development process, in general, goes through several stages, including project management, system requirements analysis, architecture, and design, implementation and testing. On each of these stages it cannot be separated from the process of capturing, storing, transferring and sharing knowledge between the developer and the stakeholders. This systematic review aimed to analyze the knowledge management process in the software development process. In this systematic review we used the Preferred Reporting Item for Systematic Review and Meta-Analysis (PRISMA) guidelines. We reviewed twenty-two works of literature from several publisher sources and analyzed the knowledge management process in software development. The analysis revealed that the knowledge management process has an essential role in improving the quality of software developed. As a conclusion of this systematic review, knowledge management is one of the factors in determining software quality.

Keywords: Software Development, Knowledge Management Process, PRISMA.

INTRODUCTION

The software development process can sometimes be very complex because it is influenced by the size of the user, the complexity of the product, and the requirements that must be met. The complexity of the development can be somehow easy if the entire software team understands and follows the process. Kraeling and Tania revealed some of the processes that must be followed in software development, which are project management, system needs analysis, architecture and design, implementation, and testing [1].

To be able to understand and follow the process, the whole software developer teams are inseparable from the knowledge management process, which is the capture process, which means the process of extracting expertise and knowledge from experts [2], storage which means storing knowledge from experts and making it accessible or reused [3], sharing which means the process of spreading new knowledge [4], while the transfer of knowledge according to Van Den Hooff and De Ridder (2004) means the process of sharing complex knowledge in a particular direction from one place, person or owner to another [5]. According to Szulanski (2000), knowledge transfer has two main components, namely the source that provides knowledge and the recipient who acquires it, knowledge transfer not only shares knowledge but also requires the application and absorption of knowledge received in a new context, so knowledge transfer results in assimilation new knowledge [5].

The quality of a software product is determined by the quality of the process that the software development team has followed. This raises the question of how knowledge management processes support or contribute to producing high-quality software products or according to the needs of stakeholders. To answer this question, we present a systematic review that identifies or analyzes the role of the knowledge management process in improving software development quality. We want to know how the knowledge management process approach supports the software development process and specifically which processes (capture, storage, sharing and transfer) support it well.

The contribution of this paper is a comprehensive review of the knowledge management process approach to software development developed in the last five years. The main aspect of this systematic review is identifying the knowledge management process in published software development in reputable journals. We identify the evidence from the study by considering the four knowledge management process variables in software development.

The rest of this paper is organized as follows. In chapter 2, we provide an overview of basic concepts of the software development process and knowledge management process. Chapter 3 presents the objectives and research questions. In chapter 4, we make a specific detail about the research methods used in preparing this systematic review. We provide details related to our money activities at each of these stages. In chapter 5, we present the results of a quality assessment of the literature review we chose and we use to answer the research questions we have set in the previous chapter. In chapter 6, we discuss the implications of this literature review in the software development process, and in chapter 7, we draw conclusions.

LITERATURE REVIEW

Software Development Process and Knowledge Management Process

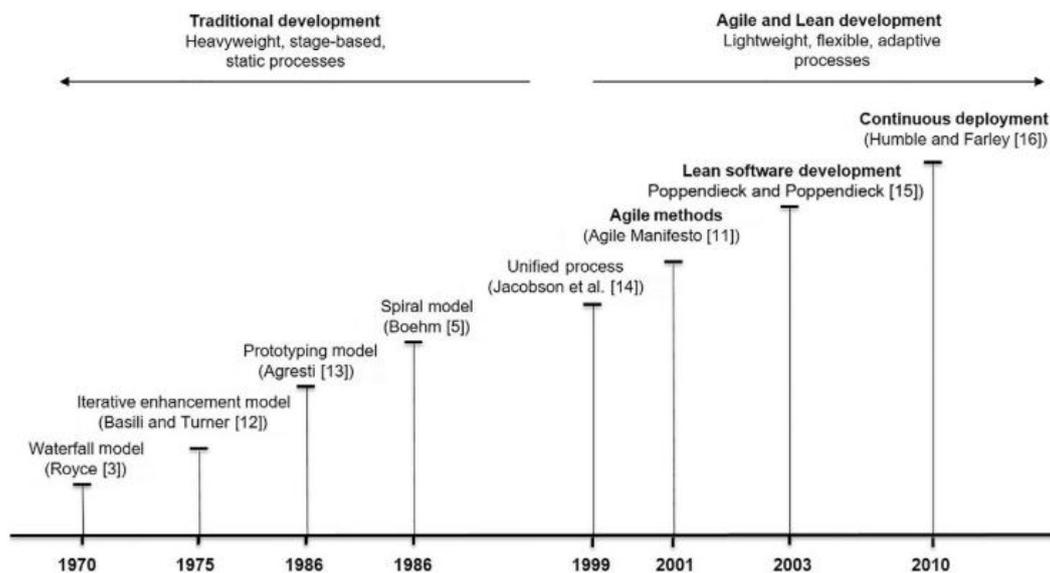
Quality management and Software Development Process

Acceptance of software quality is a match of values that can be accepted by vendors who execute the development phase with customers who outsource the work. Quality management from the perspective of the software development process consists of two considerations, namely quality control and quality assurance. Quality control as an observation technique and activity used to meet quality requirements. While quality assurance as a planned activity,

systematically implemented according to a quality system so that the quality requirements for software products can be met. In the context of the process, quality control is supported by tools and competencies of people in supporting and carrying out activities. While quality assurance takes place, the software process must be engineered through appropriate activities [6].

Group tasks in a software development process are related to the achievement of meaningful sub-goals, the use of tools that contain efforts, and quality standardization for a work system. Decomposition of objectives in a process is seen in terms of the succession of activities carried out by defining work systems, namely phases, work systems that are abstractions of processes and tasks that are primitive of a process. The phase can be identified with professional responsibility whereas the work system can be partially identified with technical objectives and fulfillment of professional responsibilities. Tasks can be identified by the process of responsibility of professionals who play a role in the development of the software. In the software process, before engaging in the software design process, we need to know the requirements and how to perform tasks in the software process. So Zope (2015) validated the software process as a good interleave between software design, software design process and software construction process [6].

The software development process gradually evolves from weights, stage-based processes, which are built on the assumptions of the requirements and divides the development process into different sequential phases based on these requirements, such as the waterfall model, prototyping, etc. As seen in Figure 1 below:



source : [7]

Figure 1. Contributions to the software development process for decades

Figure 1 illustrates a process-centric model that is considered to be deterministic and iterative which and this model has lead to a flexible approach, which emphasizes the ongoing software evolution of function, flexibility, speed, face-to-face communication, customer involvement and few artifacts [7]. The evolution of the process model occurs to fulfill the

goal of software quality. Software quality implies and states stakeholders' various needs and their values. Characteristics of software behavior is a function of system results, the system's impact on stakeholders, a measure of the level of customer satisfaction and a measure of the ability of the system to enable users to complete tasks. Everything in the system that contributes to the creation of value to stakeholders and the value carrier is considered as the quality of the system or software. Stakeholder value arises from a series of quality solutions such as function, integrity, performance, security, usability, business sustainability, response time, regulatory compliance, reuse, and others.

Knowledge Management Process

In a knowledge-based society, people are expected to not only have access to information and knowledge but be able to find value and create new information and knowledge quickly. They are also expected to communicate this to others, and work productively and collaborate with others. The process of sharing and communicating knowledge will help people to be creative and innovative [8]. Likewise in the software development process, the process of finding, organizing, transferring and using information becomes vital in achieving software quality, which is the essence of knowledge management. Knowledge management by Lee (2001) is defined as the process of capturing, storing, sharing and using knowledge.

According to the literature, various researchers have introduced knowledge management processes differently. DeLong (1997) revealed that knowledge management processes consist of the process of capturing knowledge, transferring knowledge, and applying knowledge. Whereas Probst et al. (2000) claimed that the knowledge management process includes identification of knowledge, sharing of knowledge, disseminating knowledge, applying knowledge and storing knowledge. Tiwana (2000) also revealed that the knowledge management process includes the acquisition/creation of knowledge, sharing/dissemination of knowledge and utilization of knowledge [9].

Based on a literature review of knowledge management, it can be concluded that the process that occurred in knowledge management is the process of capture, storage, and reuse, transfer, and sharing of knowledge.

RESEARCH METHODS

The overall aim of this literature review is to identify a knowledge management process approach that can support the software development process. Therefore, we formulated the research questions (RQ) as follows:

- RQ1: Does the knowledge management process (capture, storage, and reuse, transfer and sharing) support the software development process?
- RQ2: Which knowledge management process supports software development process the most?
- RQ3: What activities in the software development process can be supported by the knowledge management process approach?

In RQ1, we identified a knowledge management process approach that supports the software development process that we obtained from a systematic review. This information helps us determine the knowledge management process support that is most used in the software

development process to answer RQ2. The results of the identification list supported by our knowledge management process are used to answer RQ3. We hope that with these three RQs, readers will get insights related to knowledge management process support in the software development process.

We conducted this research as a systematic review which we compiled using the PRISMA guidelines [10]. The PRISMA Guidelines provide twenty-seven items that need to be considered in preparing a systematic review.

The literature we reviewed as material for this systematic review is the literature that discusses knowledge management processes in the software development process. We apply the eligibility criteria in the literature that we will review, namely the literature published by Science Direct, IEEE, Emerald, and Sage. We only select literature published in 2015 or more to ensure the up-to-date study that we will review. We limit the type of literature that is only literature in the form of journals and proceedings. In searching for literature we use the following keywords in each publisher source:

- “Knowledge management” & “ Software development”
- “Knowledge management” & “ Software engineering”
- “KM” & “Software development”
- “KM” & “Software engineering”

The literature collected from the search process will be filtered several times. In the first filter, we eliminate all literature from search sources by looking at the suitability of the keywords we have set previously. In the second screening, we eliminate duplicate literature. Next, we ensure the relevance of literature by reading abstracts and skimming the contents. The process of screening literature is carried out independently by each group member of two people.

After the literature is obtained, we conducted a review of the written material that will be used as a reference in writing a systematic review. As a result of this review, we posted it in a summary that lists the knowledge management process along with the activities of each literature. The process of selecting data inserted in the matrix was carried out based on the results of mutual agreement between group members. In this systematic review, we focus on activities that are included in the knowledge management process revealed by literature sources. The process of validation of reading material was carried out using a peer review method conducted by each group member to the reading material read by other group members. To minimize errors in interpretation each reviewer crosschecks the results of the review of other group members. Then the reviewer mapped some possible differences in the results of the review obtained from the literature. The main measure (primary measure) used in this systematic review is the activities in the software development process that reflect the knowledge management process.

FINDINGS AND DISCUSSION

Our search of the ScienceDirect, IEEE, Emerald, and Sage databases resulted in a total of 754 citations. However, from 754 existing literature, only 56 literature whose titles have relevance to the keywords that have been determined, most of the literature does not focus on

the software development process. From 56 literature, we made selection by removing duplicate citations. It turned out to be found one duplicate literature so that there were 55 literature left. After that, we proceed with another selecting process by reading abstracts and skimming the contents that fit or are relevant to the purpose of systematic review. In the final stage, we obtained 25 literature that met the requirements. We made these results as a reference for conducting a systematic review.

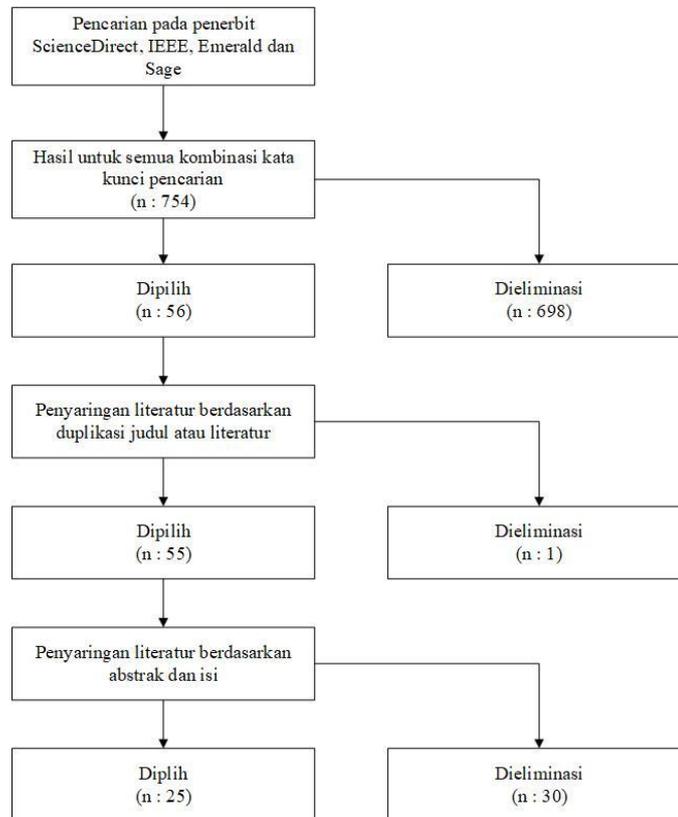


Figure 2. Literature Selection Flow

The characteristics of the literature that we have received are literature that is following the topic, which discusses the knowledge management process in the software development process. We accept types of journal literature and proceedings published in 2015 or more. In the validation process we set the control variables namely knowledge capture, knowledge storage, and reuse, knowledge sharing and transfer. From these three variables, we made a list of literature that mentioned the knowledge management process support, which we then used to answer RQ1. The results of the knowledge management process identification in the literature are shown in table 2 below:

Tabel 2.

The list of knowledge management process that supports the software development process

Literature	Knowledge capture	Knowledge storage and reuse	Knowledge sharing and transfer	Knowledge management process
[11]	No	No	Yes	Knowledge transfer in development and operational (DevOps)

Literature	Knowledge capture	Knowledge storage and reuse	Knowledge sharing and transfer	Knowledge management process
				Software development plays a role as a way to integrate development, delivery, and efficient operations in order to support and improve software quality.
[12]	No	No	Yes	Knowledge transfer plays a role in increasing effectiveness in communication and collaboration activities with stakeholders to improve the quality of the software. It manages stakeholder knowledge.
[13]	No	Yes	No	Knowledge storage and reuse play a role in increasing the effectiveness and efficiency of software maintenance by storing documentation and maintenance experience codified in a database.
[14]	No	Yes	No	Knowledge storage and reuse play a role in expanding the knowledge lost when developing software that uses agile process models.
[15]	Yes`	Yes	Yes	Knowledge capture, storage and reuse, transfer, and sharing play a role in capturing, sharing and documenting an architecture software that has implications for reducing maintenance costs and increasing system understanding.
[16]	Yes	Yes	Yes	Knowledge capture, storage and reuse, transfer, and sharing play a role in capturing, maintaining, sharing and reusing the knowledge of architecture software in the software development process.
[17]	No	Yes	No	Knowledge storage and reuse are necessary to document tacit knowledge from open-source-based software development projects. In the open-source software development project some of the contributors are volunteers with a duration of involvement, which is mostly uncertain. This documentation provides an opportunity for tacit knowledge to be reused in the future by contributors to the next open source project.

Literature	Knowledge capture	Knowledge storage and reuse	Knowledge sharing and transfer	Knowledge management process
[18]	Yes	Yes	No	<i>Knowledge capture , storage and reuse</i> digunakan sebagai strategi untuk menangkap dan menyimpan pengetahuan <i>tacit</i> dalam pengembangan perangkat lunak agar proyek pengembangan perangkat lunak No mengalami kegagalan.
[19]	No	No	Yes	Knowledge sharing and transfer play a role in increasing the effectiveness of software practice processes. However, the increase is also probably influenced by the culture that exists in the organization.
[20]	No	No	Yes	Knowledge sharing and transfer play a role in the communication process in software development, which is developed globally.
[21]	Yes	No	No	Knowledge capture plays a role in the process of capturing the tacit knowledge of the experts involved in software development projects.
[22]	Yes	No	No	Knowledge capture plays a role in the process of capturing tacit and explicit knowledge in the process of determining system requirement specifications so that stakeholder needs can be well defined.
[23]	Yes	Yes	No	Knowledge capture, storage, and reuse play a role in capturing and documenting knowledge generated in the testing process, thus facilitating the process of reusing knowledge at the next software testing stage.
[24]	Yes	Yes	No	Knowledge capture, storage, and reuse play a role in capturing and storing knowledge in the software development project process so that it can be reused in subsequent projects.
[25]	Yes	No	Yes	Knowledge capture, sharing, and transfer play a role in capturing and transferring knowledge in the communication process between end-users and software developers.

Literature	Knowledge capture	Knowledge storage and reuse	Knowledge sharing and transfer	Knowledge management process
[26]	No	No	Yes	Knowledge sharing and transfer in the development of open source-based software plays a role in the process of knowledge transfer between the software development team. The process creates a knowledge network that supports software development.
[27]	Yes	No	Yes	Knowledge capture, sharing, and transfer of software development projects play a role in detecting problems in collaborative communication between development teams so that they can support the success of software development projects.
[28]	No	No	Yes	Knowledge sharing and transfer in software development play a role in the exploration process of various skills and opportunities as well as potential in software development.
[29]	No	Yes	No	Knowledge storage and reuse in software development projects play a role in bridging the gap of understanding in software development projects. The author reveals that knowledge storage supports transfers in the system sale process.
[30]	No	No	Yes	Knowledge sharing and transfer play a role in the process of sharing experiences in problem-solving efforts that occur in the reuse of open source software
[31]	No	Yes	Yes	Knowledge storage, sharing, and transfer play a role in the process of collaborative communication practices in the software development process.
[32]	Yes	No	No	Knowledge capture plays a role in the aspect of knowledge acquisition to continuously improve the quality of software products.

Seven literature [11] [12] [19] [20] [26] [28] [30] revealed that there are knowledge sharing and transfer processes in the software development process. Four literature [13] [14] [17] [29] revealed that there is a knowledge storage process in the software development

process. Two literature [15] [16] revealed that knowledge capture, storage, and reuse dan transfer and sharing processes support the software development process. Three literature [18] [23] [24] revealed that knowledge capture and storage processes support the software development process. Two literature [25] [27] revealed that knowledge capture and transfer processes support the software development process. Three literature [21] [22] [32] revealed that the knowledge capture process supports the software development process. One literature [31] revealed that *knowledge storage and reuse* and *knowledge transfer and sharing* processes support the software development process. Although each literature shows differences, almost all claim that the knowledge management process supports the improvement of software quality.

Table 3. List of knowledge management processes that support the software development process the most

<i>Knowledge management process</i>	Total
<i>Capture</i>	10
<i>Storage and reuse</i>	10
<i>Sharing and transfer</i>	12

Table 3 above helps to answer RQ2. Table 3 shows that knowledge sharing and transfer support the process of software development most. The stages of software development supported by knowledge sharing and transfer are:

- Stages of communication with the role of supporting communication with stakeholders to ensure software requirements.
- The stages of design and plan with the role of supporting knowledge sharing and collaboration with software development teams
- The practical or operational phase or integration with the role of supporting communication between developers and users who operate the new software.

Based on table 2, researchers conducted a mapping of software development activities supported by a knowledge management process, which was then used to answer RQ3. The following Table 4 shows the matrix of the knowledge management process in software development.

Table 4. Matrix of knowledge management process at the software development stage.

<i>Knowledge capture</i>	<i>Knowledge storage and reuse</i>	<i>Knowledge sharing and transfer</i>
Architecture design software	Software Maintenance	Integration system, delivery system dan operational system
Software Project management	Communication	Communication
Software requirements analysis	Design/Architecture	Design/Architecture
Software Testing	Software Project	

	management	
Operational system	Software Testing	
	Software management	Project

Table 4 shows the stages in software development supported by the knowledge management process.

CONCLUSION AND SUGESTION

This paper highlights the existence of knowledge capture, knowledge storage, and reuse, and knowledge sharing and transfer processes in the software development process although only some of the literature reveals all three processes. Table 4 reveals the stages of software development supported by the knowledge management process so that it can provide notes to software developers to pay more attention to the knowledge process to encourage improvement in the quality of software developed.

The purpose of this systematic review is to find out the knowledge management process support for software development. Knowledge management process is identified based on three variables, namely knowledge capture, knowledge storage, and reuse, and knowledge sharing and transfer. By reviewing this literature, it is possible to identify three questions:

- RQ1: Do knowledge management process (capture, storage, and reuse, transfer and sharing) support the software development process?
- RQ2: Which Knowledge management process supports software development process the most?
- RQ3: What kind of activities in the *software development process* can be supported by *knowledge management process*?

Regarding RQ1, based on the literature identification results, it is found that each literature reveals the support of knowledge management processes, although only some contain all three processes. Thus, it can be concluded that software development requires knowledge management.

Whereas for RQ2, Table 3 highlights the number of knowledge management processes discussed in the literature, so it can be concluded that most studies discuss knowledge sharing and transfer. This shows that there are opportunities to expand research that discusses knowledge capture and knowledge sharing and transfer processes.

As for RQ3, Table 4 in this paper shows the stages of software development supported by the knowledge management process, so software developers need to increase the effectiveness and efficiency of knowledge management processes to encourage improvements in software quality. Thus, it can be concluded that the knowledge management process can be a factor in software quality.

REFERENCE

- [1] J. McDermid and P. Rook, *Software development process models*, Second Edi. Elsevier Inc., 2019.
- [2] D. Anagnostakis *et al.*, "Knowledge Capture in CMM Inspection Planning: Barriers

- and Challenges,” *Procedia CIRP*, vol. 52, pp. 216–221, 2016.
- [3] M. B. C. Zouari and S. B. D. Dakhli, “A multi-faceted analysis of knowledge management systems,” *Procedia Comput. Sci.*, vol. 138, pp. 646–654, 2018.
- [4] Z. Zheng, Y. Huyan, H. Li, S. Sun, and Y. Xu, “Quality assurance mechanisms for digital forensic investigations: knowledge sharing and the Capsule of Digital Evidence (CODE) Graeme,” *Sensors Actuators B. Chem.*, p. 127065, 2019.
- [5] J. Jie and A. A. L. Ariss, “Knowledge transfer between self-initiated expatriates and their organizations : Research propositions for managing SIEs,” *Int. Bus. Rev.*, no. June 2018, p. 101634, 2019.
- [6] N. R. Zope, K. V. Nori, A. Kumar, D. S. Lokku, S. Natarajan, and P. V. Nistala, *Quality management and software process engineering*, vol. 4. Elsevier Inc., 2015.
- [7] P. Rodríguez, M. Mäntylä, M. Oivo, L. E. Lwakatare, P. Seppänen, and P. Kuvaja, “Advances in Using Agile and Lean Processes for Software Development,” *Adv. Comput.*, vol. 113, pp. 135–224, 2019.
- [8] M. Nazim and B. Mukherjee, “An Introduction to Knowledge Management,” *Knowl. Manag. Libr.*, pp. 1–26, 2016.
- [9] M. Al-Emran, V. Mezhuyev, A. Kamaludin, and K. Shaalan, “The impact of knowledge management processes on information systems: A systematic review,” *Int. J. Inf. Manage.*, vol. 43, no. July, pp. 173–187, 2018.
- [10] T. Akhigbe, A. Zolnourian, and D. Bulters, “Compliance of systematic reviews articles in brain arteriovenous malformation with PRISMA statement guidelines: Review of literature,” *J. Clin. Neurosci.*, vol. 39, pp. 45–48, 2017.
- [11] R. Colomo-palacios, E. Fernandes, P. Soto-acosta, and X. Larrucea, “A case analysis of enabling continuous software deployment through knowledge management,” *Int. J. Inf. Manage.*, no. November, pp. 0–1, 2017.
- [12] A. Khosravi, A. Razak, C. Hussin, and M. Nilashi, “Toward Software Quality Enhancement by Customer Knowledge Management in Software Companies,” *Telemat. Informatics*, 2017.
- [13] J. B. De Vasconcelos, C. Kimble, P. Carreteiro, and Á. Rocha, “The application of knowledge management to software evolution,” *Int. J. Inf. Manage.*, 2016.
- [14] G. Borrego and A. L. Mor, “Towards a reduction in architectural knowledge vaporization during agile global software development,” *Inf. Softw. Technol.*, 2019.
- [15] R. Capilla, A. Jansen, A. Tang, P. Avgeriou, and M. A. Babar, “10 years of Software Architecture Knowledge Management: Practice and Future,” *J. Syst. Softw.*, 2015.
- [16] R. Weinreich and I. Groher, “Software architecture knowledge management approaches and their support for knowledge management activities: A systematic literature review,” *Inf. Softw. Technol.*, vol. 80, pp. 265–286, 2016.
- [17] M. Rashid, P. M. Clarke, and R. V. O. Connor, “International Journal of Information Management A systematic examination of knowledge loss in open source software projects,” *Int. J. Inf. Manage.*, vol. 46, no. November 2018, pp. 104–123, 2019.
- [18] E. S. Mtsweni and N. Maveterra, “Issues Affecting Application of Tacit Knowledge within Software Issues Affecting Application of Tacit Knowledge within Software

- Development Project Development Project,” *Procedia Comput. Sci.*, vol. 138, pp. 843–850, 2018.
- [19] J. Lee, Y. Shiue, and C. Chen, “Computers in Human Behavior Examining the impacts of organizational culture and top management support of knowledge sharing on the success of software process improvement,” *Comput. Human Behav.*, vol. 54, pp. 462–474, 2016.
- [20] M. Zahedi, M. Shahin, and M. Ali, “International Journal of Information Management A systematic review of knowledge sharing challenges and practices in global software development,” *Int. J. Inf. Manage.*, vol. 36, no. 6, pp. 995–1019, 2016.
- [21] J. Patalas-maliszewska, “Knowledge Network for the Development of Software Projects (KnowNetSoft) Knowledge Knowledge Network Network for for the the Development Development of of Software Software Projects Projects (KnowNetSoft) (KnowNetSoft) Knowledge Network for the Deve,” *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 776–781, 2018.
- [22] E. S. M, O. B. S, and A. S. A, “Knowledge meaning and management in requirements engineering,” *Int. J. Inf. Manage.*, vol. 37, no. 3, pp. 155–161, 2017.
- [23] É. Ferreira, D. Souza, R. De Almeida, and N. L. Vijaykumar, “Knowledge management initiatives in software testing : A mapping study,” 2014.
- [24] S. M. Mitchell and C. B. Seaman, “Could removal of project-level knowledge flow obstacles contribute to software process improvement ? A study of software engineer perceptions,” *Inf. Softw. Technol.*, vol. 72, pp. 151–170, 2016.
- [25] M. Zahedi and M. A. Babar, “Why does site visit matter in global software development: A knowledge-based perspective,” *Inf. Softw. Technol.*, 2016.
- [26] S. Kambiz, E. Turkina, and T. Burger-helmchen, “Knowledge management in OSS communities : Relationship between dense and sparse network structures,” *Int. J. Inf. Manage.*, vol. 38, no. 1, pp. 167–174, 2018.
- [27] A. Masuda, C. Morimoto, T. Matsuodani, and K. Tsuda, “Construction of the Collaboration Skills Knowledge in Software Development,” *Procedia - Procedia Comput. Sci.*, vol. 96, pp. 1129–1136, 2016.
- [28] S. Ghobadi, “What drives knowledge sharing in software development teams: A literature review and classification framework,” *Inf. Manag.*, vol. 52, no. 1, pp. 82–97, 2015.
- [29] P. Savolainen and J. J. Ahonen, “Knowledge lost: Challenges in changing project manager between sales and implementation in software projects,” *Int. J. Proj. Manag.*, vol. 33, no. 1, pp. 92–102, 2015.
- [30] H. M. Koo and I. Y. Ko, “Construction and utilization of problem-solving knowledge in open source software environments,” *J. Syst. Softw.*, vol. 131, pp. 402–418, 2017.
- [31] A. Menolli, M. A. Cunha, S. Reinehr, and A. Malucelli, “‘Old’ theories, ‘new’ technologies: Understanding knowledge sharing and learning in Brazilian software development companies,” *Inf. Softw. Technol.*, vol. 58, pp. 289–303, 2015.

[32] Z. Micić and M. Blagojević, “Knowledge acquisition in information technology and software engineering towards excellence of information systems based on the standardisation platform,” *Comput. Stand. Interfaces*, vol. 44, pp. 1–17, 2016.