

Exploring Knowledge Management From a Software Engineering Perspective

Dilyan Georgiev

Faculty of Mathematics and Informatics, Sofia University St. Kliment Ohridski, Bulgaria

diljang@uni-sofia.bg

Abstract: *Context:* The knowledge domain of software engineering (SE) gradually expands due to fast emerging technologies and complex organisational processes. The software industry represents a special focus of interest from a Knowledge Management (KM) perspective, as it is characterized by high employee turnover, a high level of outsourced projects, and an accelerated pace of technology innovations. Knowledge management can substantially influence SE knowledge's life cycle, contributing to better structuring of organisational processes of knowledge creation, knowledge transfer, and knowledge sharing. Reflecting on the trends for SE automation with artificial intelligence tools, KM approaches can enable companies and professionals to reconsider the future of SE work. *Purpose:* The present research aims to identify and investigate how implementing knowledge management principles and processes within the software engineering domain can facilitate the future transformation of software-related jobs. Based on a conceptual model, built upon the literature review, the paper analyses the quantitative data, collected from 91 software experts. Evaluating the impact of both organizational and domain types of SE knowledge, the discussion evaluates the unique value-adding SE processes along with models of how to explore AI applications to automate the SE domain. *Results:* The outcomes of the analysis identify the models of structuring knowledge management processes in SE both within the technological domain and organisational layers. The discussion section establishes suitable KM methods and approaches for supporting SE job transformation. It identifies the main knowledge management integration practices, demonstrating how combining KM with AI-based innovations may change the SE knowledge flow, improving substantially SE activities like onboarding new team members, monitoring software documentation, version control, and error tracking.

Keywords: Software engineering, Knowledge management, Software process improvement, Knowledge sharing, AI, software industry

1. Introduction

The knowledge domain of Software engineering (SE) is fast enlarging, influenced by various dynamic factors due to technological novelties, organisational and management practices. Responding to the need to build reliable, efficient, and scalable software systems, software companies have adopted a variety of SE methodologies, tools, and techniques (Kiv et al., 2018). The software engineering work is reflected often as a “meta-industry” supporting the digitalization and digital transformation of all other economic sectors and activities. It is investigated from various knowledge management perspectives, recognized as “knowledge-intensive processes” (Bjørnson et al, 2008), both on individual and organizational level. Knowledge management can substantially support the SE knowledge's life cycle, contributing to better structuring of organisational processes of knowledge creation, knowledge transfer, and knowledge sharing. SE professionals are expected to access and use knowledge from various sources, including design patterns, programming languages, software architecture, and best practices. Further, the process of software development generates enormous volumes of data and knowledge, that must be further captured, codified, and re-used within the organisation. However, the recent advancements in Artificial Intelligence (AI), have the potential to revolutionize many aspects of modern society, including software development processes and the software engineering profession.

The rapid pace of technological advances requires software companies to adopt emerging approaches, applications, and tools for preserving their competitiveness and market position. This process of aligning to the latest technology developments is time-consuming and demanding, both for organizations and SE professionals, who must stay up-to-date and deliver high-quality software. Moreover, the SE is a highly collaborative process and teams need to cooperate while managing real-time large volumes of data and knowledge, sharing code libraries, documentation, and analysing user feedback. At the same time, software development is highly specialized, and every employee can develop unique knowledge and expertise, which can be hardly captured and codified by teams or by companies. In principle, this expertise is processed within various Information Systems (IS), making a “clear distinction between data, information, and knowledge” (Baydoun et al, 2021), and applying customisations approaches. Specific knowledge-sharing activities are designed to capture and retain knowledge and to prevent the loss of valuable expertise, and companies recognize that “knowledge sharing influences team performance, creativity, and climate” (Ahmad et al, 2019). At the same time, the fast-emerging Artificial intelligence tools promise to revolutionize the SE processes in organisations and to further automate the process of software development. Therefore, companies need to better understand the specific and unique

SE expertise and knowledge management practices, that can be assisted by AI. It is already admitted that AI tools can be used to improve knowledge sharing in various ways, including enhancing search capabilities, enabling personalized content delivery, and automating knowledge-intensive tasks.

The present research aims to identify and investigate how implementing knowledge management principles and processes within the software engineering (SE) domain can facilitate further software-related job transformation.

The paper is structured as follows. Based on a preliminary background analysis, a conceptual model is defined, combining organizational and domain types of SE knowledge. It serves as a starting point for evaluating unique value-adding SE processes and knowledge. Next, the analysis of the responses, provided by 91 software experts, examine the trends and actual expectation between different companies and software professionals. The obtained results help to illustrate the need for structuring knowledge management processes within the technological domain and between organisational layers. In the end, the discussion part evaluates the suitable KM methods and approaches for supporting SE job transformation and identifies the main aspects that AI technologies can support knowledge management integration in SE practices.

2. Background

Software development is a knowledge-intensive domain, thoroughly investigated from a KM perspective in several diverse systematic reviews (Bjørnson & Dingsøyr, 2008; Ouriques et al., 2019, Mora et al., 2019 and others). The software development process along with the SE life cycle is context-dependent, the application of knowledge requires additional preparation, and stakeholders need to be aware of the available explicit and tacit knowledge (Ouriques et al, 2019). Agile concepts are a means of tailoring software development to draw the development team's attention, strengthen the decision-making process, increase inter-team coordination, and build collective-code ownership (Kiv et al, 2018). To generate new ideas that could support the product, expand its scope, and implement its functionalities according to stakeholders' expectations, creation, storage, transfer, and retrieval activities must be introduced. To ensure product quality and simplify and expedite integration processes with the production environment, frameworks such as DevOps (Development and Operations) are available for the product delivery (Céspedes et al, 2017). These goals align with customer expectations of building trust, bringing satisfaction, and establishing the belief that "working software is the primary measure of progress" (Mora et al, 2019).

A preliminary research (Georgiev, 2023) analyses 16 systematic reviews of different knowledge domains, outlining the key topics between SE and KM. To define the SE Knowledge domain, thorough research is made, which summarises the reviews and their main topics. Thus, by comparing the main concepts, a joint knowledge domain emerges, where subdomains are identified and highlighted. Those subdomains include Software Process Improvement, KM in large projects on a global scale, and KM factors regarding Agile methodologies and Knowledge Management Systems.

Based on the thorough analysis of the literature, there are identified four main knowledge domains in software companies, critical for their success. These knowledge domains include technocratic, behavioural, product, and process types of knowledge:

- *Technocratic knowledge* - several studies highlight the necessity of adopting a more technocratic approach to knowledge management (KM). This approach involves utilizing cognitive analysis, closely examining communication within virtual reality, employing event modelling, and clustering data based on common features (Osorio et al, 2020). In project management, this is crucial as codification and documentation play a vital role in monitoring (Ayarza et al, 2020). Knowledge modelling, which focuses on applying conceptual and computational models to extract functions or process clustered data, also plays a significant role. Managing perceived complexity and risks is essential in utilizing knowledge artifacts. Therefore, many aspects of knowledge management can be classified under this technocratic approach.
- *Behavioural knowledge* is a critical domain in software companies that involves team modelling to identify specific behaviour types within a team and patterns in larger projects. This approach prioritizes human factors, which are introduced effectively in the software engineering process to improve it via education and motivation. Agile frameworks such as Large-Scale Scrum (LeSS) (Steghöfer et al, 2019) and Scaled Agile Framework (SAFe) (Putta et al, 2018) have been successful in building compact and effective teams that quickly deliver value to clients' expectations. However, the main driver of the agile transformation is the agile coach who builds teams, creates guidelines, sets

goals and roadmaps, and leads the team toward self-organization. The success of the agile transformation is evaluated based on shared product vision, shared responsibilities, shared knowledge, feedback, and ownership, while failure factors include lack of middle management support, barriers to the production environment, excessive control by higher management, and lack of understanding from stakeholders.

- *Product knowledge* is used to transform software systems "into easily accessible, well described, and interoperable, modular services" (Huang et al, 2008). Descriptive languages provide a machine-readable domain specification that can be enhanced by introducing metadata and conflict-detecting mechanisms through a three-step analytic process for validation, verification, and performance metrics. This approach is used to transform legacy systems into easily accessible, well-described, and interoperable modular services. Languages like UML, VEL, (Bagnato et al, 2019), and WADL support different development designs, including Quality-driven Architecture Design (QDAD) (Franch et al, 2019), Quality-Aware Rapid Software Development Design (Q-Rapids), Transformation models in dynamic environments (Trieflinger et al, 2021), and older pipeline development models like Product Line Use case modelling for Systems and Software engineering, described by Ahmed et al (2008).
- *Process knowledge* refers to the knowledge of the internal processes and procedures that a software company uses to deliver its products or services. These solutions vary, with some focusing on data handling, while others concentrate on monitoring changes and product engineering. In the initial stages of product development, the main goal is to clarify requirements and develop a solution based on past experiences. The portfolio-driven development model, an advanced model of Agile Product Line Engineering (APLE), demands a more agile and rapid approach that involves "a collection of projects, programs, and other operations for achieving the business goals." (Hayashi et al, 2020). With this, collection managers can provide more precise estimations, especially when working with fixed resources.

Each of these knowledge domains can be critical for employees learning and growth. However, the relative importance of each domain may vary depending on the company's size, structure, and experience. For example, a start-up software company may prioritize technocratic knowledge to develop a new product. In contrast, a large IT services company may prioritize process knowledge to ensure efficient development procedures.

3. Study Design and Implementation

To investigate the current situation in the SE knowledge domain, a short survey is prepared and shared among SE professionals in Bulgaria. The survey is divided into four parts, each representing questions that aim to gather information about employees' approaches to software processes, team methodologies and established traditions, organizational culture, and solutions they are working on. The questions are classified into the four knowledge domains mentioned above. The first part of the survey gathers information about employees' work such as workload, activities within the working process, and overall job satisfaction. The second part focuses on teamwork, collaboration, communication, and group effectiveness. The third part gathers information about organizational culture, exploring leadership, innovation, and employee engagement. The fourth and final part focuses on solutions the employees are working on, including questions related to the product and process domains, and their familiarity with the product, their involvement in the software lifecycle, and the efficiency of the development process.

The survey is circulated in the months of January and February 2023, collecting 91 responses among SE professionals. The analysis of the results considers employee experience, company size, and organizational structure. Average values are calculated and compared to the overall ones. This process helps to identify trends and potential problems.

3.1 The Respondent's Profiles and Expertise

The analysis of the respondents' profiles is presented in Figure 1, exploring their general SE expertise. It is important to mention that almost 90% of the respondents work in full-time job positions. Considering the SE job expertise, about 73% of the respondents work as software developers, more than 25% have some management background as product/project management, scrum master or team lead and 13% are software architects, 11% are quality assurance (QA). Business and marketing positions (7%) and HR/Recruitment experts are 5%. It should be mentioned that almost 42% share two or more job positions.

Considering the experience in the SE domain, more than 53% of the respondents have substantial experience and SE expertise (20% more than 10 years, and 33% more than 5 years). It is important to mention that only 14%

are in junior positions with less than 1 year of experience. About 51% of the respondents work in big companies with more than 100 employees. About 26% work in small companies (less than 20 employees).

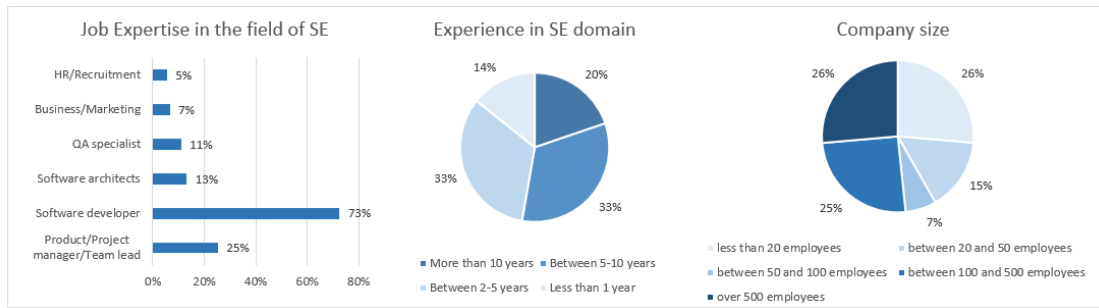


Figure 1: Respondents' Profile

The general analysis shows that the survey gathered various expertise and experiences which provides a diverse data sample. This allows critical points to be identified in terms of what motivates respondents' growth, how they react to changes within their work, and how companies can support their work. It is essential to consider the following factors when investigating further the results:

- company size – the number of employees, volumes of data, and knowledge infrastructure contribute to the KM solutions;
- project complexity – how the scope of product knowledge and architectural solutions influence technological progress;
- team size – how the bonding experiences and teamwork are introduced in organisational culture;
- experience – how different professional backgrounds could diversify the knowledge base;
- organizational structure – how different structure types correspond to the communication channels within the company.

These factors impact knowledge types and necessitate strategies to mitigate information noise and redundant procedures and improve the accessibility and efficiency of KM processes.

4. Discussion

4.1 Analysis of the Quantitative Results

The main part of the survey explores the four main knowledge domains, focusing on SE products and processes and SE organizational aspects from technocratic and behavioural styles. Using a Likert scale (1- definitively not, 5 – definitively yes, 3 -neutral), four groups of questions are formulated.

Figure 2 presents the questions about Domain knowledge, divided into the following groups:

SE Domain – Product, exploring how requirements engineering (RE) and quality assurance (QA) affect the product's evolution and how employees react to those processes (Figure 02). Most SE professionals value the importance of a high-level quality of the software products, as well as the influence of how functionalities are described in their work as programmers.

SE Domain – Process, exploring how Continuous Integration (CI) provides a flexible approach to effective value delivery. They also analyse the communication within the processes of development and management. Experts value the adoption of different concepts to have a better knowledge base and efficient value-generating process.

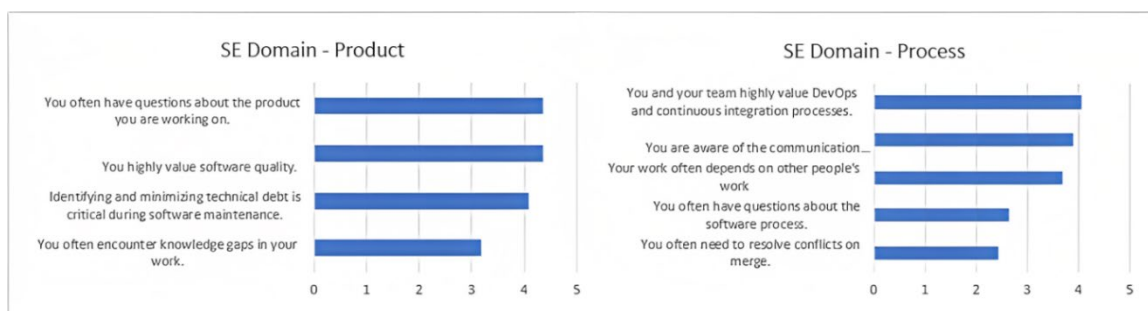


Figure 2: Product and Process Questions Quantitative Results

Figure 3 presents the questions about Organisational knowledge, also divided as follows:

SE Organisational - Technocratic, exploring product quality in terms of how respondents' colleagues perceive it as a critical process for their success. Then the relationship between organisational structure and the product is also included – experts are supportive of the paradigm which states that communications through the organisation follow the architectural style the project defines.

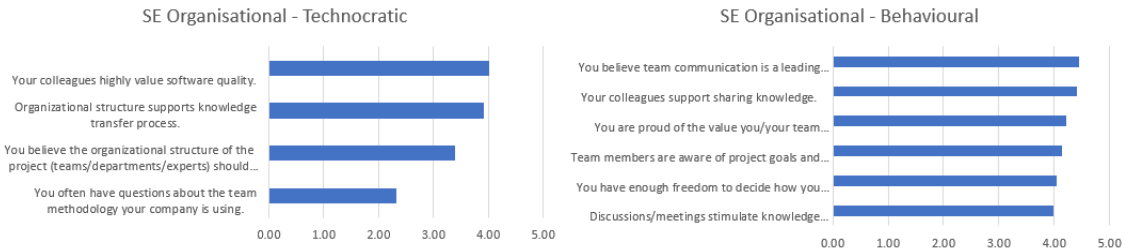


Figure 3: Technocratic and Behavioural Questions Qualitative Results

SE Organisational – Behavioural, exploring two perspectives - the first one covers trends on a local scale within the team – employees are quite supportive in terms of sharing knowledge and generating ideas for boosting efficiency. The second one measures the overall acceptance and procedures that support organisational culture – employees are rather positive about their career growth and are expecting effective approaches like mentoring and onboarding for bigger projects.

4.2 Knowledge gap Analysis

Process knowledge gaps

Most issues about the process knowledge domain stem from the software development life cycle. When aiming to create value, it is crucial to possess a clear understanding of the functionalities under development and align them with previous versions of the software. This establishes a strong interdependence among teams and team members. Over 60% of experts highlight this interdependence, which escalates the risk of conflicts, particularly in larger projects. Hence, it becomes necessary to control knowledge generation by implementing collaborative methods and groupware, while also incorporating automated procedures that synchronize not only with Version Control Systems but also with methodologies that facilitate effective value creation and modelling.

Product knowledge gaps

The problem associated with product knowledge primarily stems from its accessibility and availability. According to half of the respondents, they frequently encounter questions about the software they develop. This issue has a significant impact on the entire lifecycle from three different perspectives: software quality, technical debt, and legacy software.

Firstly, it affects software quality as it hinders the thorough testing of basic cases, edge cases, and complex end-to-end scenarios. Without adequate product knowledge, developers may struggle to identify and address potential issues effectively.

Secondly, insufficient product knowledge often results in technical debt, causing performance issues and decreased motivation among the development team. When developers lack a deep understanding of the software, they may resort to quick fixes or suboptimal solutions, leading to bad quality and low flexibility.

Lastly, legacy software often fails to provide adequate support and sufficient opportunities for professional growth. Knowledge gaps regarding such software can arise due to outdated technologies, the absence of a codification strategy, and a lack of investments.

Technocratic knowledge gaps

The three main problems addressed by questions related to technocratic knowledge are as follows:

- lack of mentoring (70%) is closely linked to Creative Organizational Learning (Sahibzada et al., 2020), which is crucial for fostering the growth of intellectual capital. Mentoring plays a vital role in guiding individuals and nurturing their skills and knowledge within an organization.

- lack of new technology knowledge (42%). Staying updated with the latest technological advancements is essential for organizations to remain competitive and adapt to changing market trends. Acquiring and integrating new technology knowledge is crucial for their success.
- lack of adequate documentation (29%). While considered a secondary knowledge source, documentation serves as a solid foundation for initiating new projects, facilitating hiring processes, and managing employee terminations. Having well-documented information ensures continuity and knowledge transfer within an organization.

In response to these challenges, the majority of respondents agree or strongly agree that organizational structures should facilitate knowledge sharing both horizontally (70%) across management layers and vertically among team members (58%). To promote effective knowledge exchange and utilize knowledge artifacts, organizations implement various methods such as brainstorming sessions, discussions, team-building activities, and mentoring-focused onboarding programs. By prioritizing mentoring over documentation, organizations recognize the value of experiential learning and interpersonal knowledge transfer.

Given the rapid increase in information volumes and overall data dynamics, knowledge management emerges as a critical supporting process, particularly for fast-growing companies. It enables efficient handling of information, facilitates knowledge sharing, and helps organizations navigate the ever-changing landscape of information and data.

Behavioural knowledge gaps

Upon analysing the results, a significant observation emerges regarding the reliance of the behavioural domain on tacit knowledge. Notably, new employees heavily depend on more experienced colleagues, including team members (75%), team leaders (48%), and managers (21%), to acquire this knowledge. The key requirements for these individuals revolve around their proactivity, motivation to grow, and effective communication skills to navigate the dynamics of communication within and between teams.

Another noteworthy gap relates to communication between different management layers. Approximately 60% of the respondents express the importance of understanding the organizational structure and how knowledge flows across different departments. To address this gap, knowledge management techniques primarily focus on stimulating proactivity and idea generation. Consequently, various initiatives are implemented to foster bonding experiences and cultivate a culture of sharing within the organization. Furthermore, this corresponds to various studies stating that knowledge sharing plays a crucial role in "moderating the relationship between transformational leadership and innovative work behaviour" (Afsar et al., 2019).

4.3 Management Perspectives

Communication serves as a fundamental pillar in the construction of knowledge infrastructure. Consequently, several factors must be considered when establishing information flows and structures, including company size, which impacts the volume of data being processed and the risk of knowledge loss. A prominent approach in this regard is Business Model Innovation (BMI), which proves advantageous for companies seeking to reconfigure the delivery and capture of value from customers (Hock-Doepgen et al., 2021). This approach not only stimulates discussions about process optimization but also raises the question of how to provide customized solutions for different technological challenges. The notion of a strong interdependence between internal knowledge management and external knowledge provision introduces new requirements for employees, such as mentoring, team bonding, workshops, and opportunities for learning and growth. These initiatives effectively bridge knowledge gaps and facilitate the process of value creation.

4.4 Implications for Future Implementation of AI

Although the survey is not focused on AI, the increased interest in AI-based applications already has its influence in different directions, having several technologies introduced to facilitate the filling of knowledge gaps for employees. While internet resources serve as a static, default solution for accessing general knowledge such as code blocks, algorithms, and methods, customized standards, working solutions, and decisions that significantly impact the knowledge process are typically provided through internal knowledge infrastructure. To support knowledge generation, companies implement various software systems, including:

- Documentation software: This software utilizes different plugins for tracking, sharing, and versioning documents.
- Collaboration software: These tools employ features like whiteboards, screen sharing, and information exchange to facilitate collaboration among employees.

- DevOps and Version Control Systems (VCS): Particularly valuable for software development, VCS systems like Git enable new experts to access, update, and manage knowledge effectively.

AI-powered tools can also automate documentation and update it automatically based on code or product changes. This saves time and reduces the workload of employees responsible for documentation maintenance.

Some additional examples such as ChatGPT provides accurate search results, enabling employees to quickly find the information they need without spending excessive time searching through documents. Furthermore, AI can analyse search data to identify knowledge gaps and areas where additional information is required.

Tools like Github CoPilot suggest code snippets, reducing the time spent on code reviews and expediting deployment. Through analysis of extensive code bases, AI can detect syntax errors, performance issues, and security vulnerabilities. It can also propose solutions, including refactoring and optimization techniques.

DeepCode is another tool that aids in error identification, performance optimization, and security vulnerability detection. It can also identify patterns in code and assist in decision-making processes.

Microsoft Power BI and Google Cloud Auto ML can identify trends and provide insights that aid in informed decision-making. This is particularly beneficial for new employees who are not yet familiar with the organization's workflows and processes, thereby facilitating knowledge sharing.

5. Conclusion

In conclusion, the questionnaire results offer valuable insights into how knowledge is organised within the software industry. They clarify the existing knowledge management challenges in software engineering. One major issue identified is the difficulty in sharing knowledge, particularly when complex tasks are involved, that require significant time and effort. These challenges can lead to workflow delays and inefficiencies. Another issue is how knowledge management tools are being used to improve overall communication both on local and global scales. Lastly, organisational culture is the main driver of how team members bond and work more efficiently together. KM should support integrating employees with the company's culture and be a major driver in establishing a creative working environment.

AI-powered tools and automated systems can enhance the productivity and efficiency of software developers. By integrating AI into their workflows, organizations can save time, reduce errors, and expedite their processes. AI can introduce practical approaches based on theoretical knowledge.

Some of the challenges mentioned above can be addressed through innovative applications. As stated in the last section, there are ideas and prototypes that have already been implemented technologically and are widely used by experts. Thus, the KM process is integrated effortlessly, and its contributions would be highly evaluated.

Finally, this research serves as a starting point for an advanced investigation process, which could identify patterns in KM from an SE perspective by adding identified factors such as company size, team size, project complexity, and experience. It is important to compare how they correspond to the organisational structure and how AI can handle different cases to optimise the common knowledge workflow.

References

- Afsar, B., Masood, M. and Umrani, W.A. (2019) The role of job crafting and knowledge sharing on the effect of transformational leadership on innovative work behavior, *Personnel Review*, Vol. 48 No. 5, pp. 1186-1208. <https://doi.org/10.1108/PR-04-2018-0133>
- Ahmad, Farhan and Karim, Muhaimin (2019) Impacts of Knowledge Sharing: A Review and Directions for Future Research (May 21, 2019). Ahmad, F., & Karim, M. Impacts of knowledge sharing: a review and directions for future research. *Journal of Workplace Learning*, 31(3), 207-230. DOI/10.1108/JWL-07-2018-0096, Available at SSRN: <https://ssrn.com/abstract=3392007>
- Ahmed F., Capretz, L. F. (2008) The software product line architecture: An empirical investigation of key process activities. In: *Information and Software Technology*, Volume 50, Issue 11, 2008, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2007.10.013> pp 1098-1113
- Ayarza, V.C., Bayona-Oré, S. (2020) Cluster Monitoring and Integration in Technology Company. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) *Trends and Applications in Software Engineering. CIMPS 2019. Advances in Intelligent Systems and Computing*, vol 1071. Springer, Cham. https://doi.org/10.1007/978-3-030-33547-2_19 pp 253-265
- Bagnato, A. et al. (2019) Showcasing Modelio and pure:variants Integration in REVaMP2 Project. In: Franch, X., Männistö, T., Martínez-Fernández, S. (eds) *Product-Focused Software Process Improvement. PROFES 2019. Lecture Notes in Computer Science*, vol 11915. Springer, Cham. https://doi.org/10.1007/978-3-030-35333-9_43 pp 590-595

- Baydoun, Ali & Ahmad, Fatima (2021) Knowledge base cloud framework for effective knowledge retention in software supply chain. *International Journal of Engineering Applied Sciences and Technology*. 6. 44-52. [10.33564/IJEAST.2021.v06i08.007](https://doi.org/10.33564/IJEAST.2021.v06i08.007).
- Bjørnson, F.O., Dingsøyr, T. (2008) Knowledge management in software engineering: A systematic review of studied concepts, findings and research methods used. *Information and Software Technology*, Volume 50, Issue 11, 2008, pp 1055-1068
- Céspedes, D., Angeleri, P., Melendez, K., Dávila, A. (2019) Software Product Quality in DevOps Contexts: A Systematic Literature Review. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) *Trends and Applications in Software Engineering*. CIMPS 2019. *Advances in Intelligent Systems and Computing*, vol 1071. pp 51–64
- Franch, X., Lopez, L., Martínez-Fernández, S., Oriol, M., Rodríguez, P., Trendowicz, A. (2019) Quality-Aware Rapid Software Development Project: The Q-Rapids Project. In: Mazzara, M., Bruel, JM., Meyer, B., Petrenko, A. (eds) *Software Technology: Methods and Tools*. TOOLS 2019. *Lecture Notes in Computer Science()*, vol 11771. Springer (2019), Cham. https://doi.org/10.1007/978-3-030-29852-4_32 pp 378–392
- Georgiev, D. (2023) Exploring software engineering knowledge domains, presented on Faculty of Mathematics and Informatics – Spring Science Session 2023, Section Software Technologies and Big Data, Department of Software Technologies
- Hayashi, K., Aoyama, M. (2020) A Portfolio-Driven Development Model and Its Management Method of Agile Product Line Engineering Applied to Automotive Software Development. In: Morisio, M., Torchiano, M., Jedlitschka, A. (eds) *Product-Focused Software Process Improvement*. PROFES 2020. *Lecture Notes in Computer Science*, vol 12562. Springer (2020), Cham. https://doi.org/10.1007/978-3-030-64148-1_6 pp 88–105
- Hock-Doepgen, M., Clauss T., Kraus S., Cheng Ch., (2021) Knowledge management capabilities and organizational risk-taking for business model innovation in SMEs, *Journal of Business Research*, Volume 130, 2021, Pages 683-697, ISSN 0148-2963, <https://doi.org/10.1016/j.jbusres.2019.12.001>
- Huang, S.M., Chu, Y.T, Li, S.H., Yen, D.C. (2008) Enhancing conflict detecting mechanism for Web Services composition: A business process flow model transformation approach. In: *Information and Software Technology*, Volume 50, Issue 11, 2008, ISSN 0950-5849, <https://doi.org/10.1016/j.infsof.2007.10.014> pp 1069-1087
- Kiv, S., Heng, S., Kolp, M., Wautelet, Y. (2018) Agile Manifesto and Practices Selection for Tailoring Software Development: A Systematic Literature Review. In: *Product-Focused Software Process Improvement*. PROFES 2018. *Lecture Notes in Computer Science()*, vol 11271. pp 12–30
- Mora, M., Wang, F., Gómez, J.M., Díaz, O. (2019) A Comparative Review on the Agile Tenets in the IT Service Management and the Software Engineering Domains. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) *Trends and Applications in Software Engineering*. CIMPS 2019. *Advances in Intelligent Systems and Computing*, vol 1071. pp 102–115
- Osorio Angel, S., Peña Pérez Negrón, A., Espinoza Valdez, A. (2020) From a Conceptual to a Computational Model of Cognitive Emotional Process for Engineering Students. In: Mejia, J., Muñoz, M., Rocha, Á., A. Calvo-Manzano, J. (eds) *Trends and Applications in Software Engineering*. CIMPS 2019. *Advances in Intelligent Systems and Computing*, vol 1071. Springer (2020), Cham. https://doi.org/10.1007/978-3-030-33547-2_14 pp 173–186
- Ouriques, R., Wnuk, K., Gorschek, T., Berntsson, R. (2019) Knowledge Management Strategies and Processes in Agile Software Development: A Systematic Literature Review. *International Journal of Software Engineering and Knowledge Engineering* (2019), Vol. 29, No. 03, pp. 345-380, 10.1142/S0218194019500153
- Putta, A., Paasivaara, M., Lassenius, C. (2018) Benefits and Challenges of Adopting the Scaled Agile Framework (SAFe): Preliminary Results from a Multivocal Literature Review. In: , et al. *Product-Focused Software Process Improvement*. PROFES 2018. *Lecture Notes in Computer Science()*, vol 11271. Springer (2018), Cham. https://doi.org/10.1007/978-3-030-03673-7_24 pp 334–351
- Sahibzada, Umar & Jianfeng, Cai & Latif, Fawad & Shah, Syed & Sahibzada, Hassam. (2020). Refuelling knowledge management processes towards organisational performance: mediating role of creative organisational learning. *Knowledge Management Research & Practice*. 21. 1-13. 10.1080/14778238.2020.1787802.
- Steghöfer, JP., Knauss, E., Horkoff, J., Wohlrab, R. (2019) Challenges of Scaled Agile for Safety-Critical Systems. In: Franch, X., Männistö, T., Martínez-Fernández, S. (eds) *Product-Focused Software Process Improvement*. PROFES 2019. *Lecture Notes in Computer Science()*, vol 11915. Springer (2019), Cham. https://doi.org/10.1007/978-3-030-35333-9_26 pp 350–366
- Trieflinger, S., Münch, J., Wagner, S., Lang, D., Roling, B. (2021) A Transformation Model for Excelling in Product Roadmapping in Dynamic and Uncertain Market Environments. In: Ardito, L., Jedlitschka, A., Morisio, M., Torchiano, M. (eds) *Product-Focused Software Process Improvement*. PROFES 2021. *Lecture Notes in Computer Science()*, vol 13126. Springer(2021), Cham. https://doi.org/10.1007/978-3-030-91452-3_9 pp 136–151