KnowEval: A Tool for Supporting Knowledge Transfer Evaluation

José Wilson Cavalcanti Ferreira Junior
Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Recife, Pernambuco
jwcfj@cin.ufpe.br

Williby Ferreira
Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Recife, Pernambuco
wsf2@cin.ufpe.br

Juliana Saraiva
Departamento de Ciências Exatas
Universidade Federal da Paraíba (UFPB)
Rio Tinto, Paraíba
julianajags@dcx.ufpb.br

Sérgio Soares
Centro de Informática
Universidade Federal de Pernambuco (UFPE)
Recife, Pernambuco
scbs@cin.ufpe.br

ABSTRACT

The evaluation of Knowledge Transfer (KT) processes is essential for promoting organizational learning, enhancing operational efficiency, and fostering innovation. However, existing tools described in the literature offer limited support to organizations in this area. In general, knowledge about these tools remains unsystematized, lacks empirical validation, and is not widely adopted in practice. This article presents an overview of the key features of KnowEval, a decision support tool designed to recommend appropriate processes for KT evaluation based on strategic input variables. Furthermore, we describe the structured and dynamic database that enables the continuous integration of new evaluation processes. Finally, we present the validation results of KnowEval, using scenarios derived from an exploratory review of the literature. Video Available on: https://zenodo.org/records/15485087

KEYWORDS

Knowledge Transfer, Process Evaluation, Web-based tool.

1 Introduction

The advancement of technological innovation and the effects of globalization have made knowledge an indispensable resource for organizational competitiveness [1]. However, the increasing complexity of organizational environments highlights the urgent need to develop effective knowledge management (KM) practices [6]. This need poses a challenge for organizations to identify and structure knowledge, thereby creating efficient conditions for its sharing among individuals, teams, and processes [2].

In this context, knowledge transfer (KT) emerges as a strategic element of KM, understood as a relational process through which knowledge, practices, and experiences are shared, disseminated, and applied within organizations [25, 27]. However, the operationalization of KT faces challenges across a wide range of application domains [7]. Among these challenges are the low prioritization of KM in decision-making processes, difficulties in aligning KT objectives with organizational strategies, and the absence of clear criteria for evaluating the effectiveness of KT [12, 14]. These limitations become even more critical in contexts such as software engineering, where technical complexity and the dynamic nature of projects demand more robust and adaptable KT approaches [23].

In software development environments, KT is often hindered by various factors, such as the tacit nature of knowledge (originating from individual experiences), high employee turnover, the scarcity of explicit knowledge (formalized and structured documentation), and the reliance on informal communication practices [11]. Additionally, the lack of KT evaluation impairs the feedback loop in development processes, hinders the mitigation of risks associated with knowledge loss, and undermines the sustainability of organizational capabilities built upon accumulated learning [19].

The evaluation of KT, therefore, represents a critical mechanism for diagnosing barriers, identifying best practices, and measuring the impact of transferred knowledge on organizational processes [12, 29]. However, this evaluation is still addressed in an incipient manner, inhibiting the establishment of systematized assessment practices that could help organizations recognize recurring failures in knowledge flow [19, 25].

On a global scale, it is observed that in the United States, evaluation methods are often limited to the number of registered patents, leading to significant losses in KT within technology-based organizations [1]. In Europe, some initiatives have attempted to incorporate broader indicators, yet the lack of consensus on evaluation parameters continues to limit the effectiveness of such efforts [8]. In Brazil, despite public policies that promote KT and technological advancement, scientific output on KT evaluation remains low, positioning the country unfavorably in comparative international studies [2]. Therefore, the evaluation of KT is relevant to various types of organizations.

This landscape highlights the limited availability of mechanisms capable of supporting organizations in the structured and systematic evaluation of KT [12]. Furthermore, empirically grounded support tools that foster practical applicability while aligning with organizational needs are rarely adopted [11, 13]. In this context, the research problem lies in the lack of tools that systematize knowledge and assist organizations in the evaluation of KT. The development of such a solution may contribute to strengthening organizational knowledge, diagnosing weaknesses in knowledge flows, encouraging the dissemination of transferable best practices across projects and teams, and promoting the continuous improvement of software development processes.

2 Background

This section broadens and supports the understanding of KT evaluation in the context of software engineering and introduces essential concepts related to decision support systems.

2.1 Knowledge Transfer Evaluation in Software Engineering

Organizational knowledge is often categorized into two forms: tacit and explicit [27]. Tacit knowledge is based on individual experience and is difficult to formalize. In contrast, explicit knowledge can be easily articulated, codified, and transmitted through documentation [28, 29]. The knowledge-intensive nature of software engineering demands even more effective management of both types of knowledge. In this context, KT in software engineering is essential to ensure that lessons learned and technical knowledge are effectively disseminated [21, 22].

The evaluation of KT in software environments requires a holistic perspective on: (i) organizational structures, as they define responsibilities, influence communication flows, and enable flexible frameworks that facilitate KT; (ii) operational processes, which must be aligned with organizational goals and well-structured to ensure the capture and dissemination of knowledge throughout the software lifecycle; and (iii) technical infrastructure, consisting of tools and systems that support KT by enabling effective knowledge flows within the organization [14, 24, 29].

2.2 Decision Support Systems

Interactive systems that assist stakeholders in making complex decisions involving large volumes of data and multiple analysis criteria are known as DSS – Decision Support Systems [3, 17]. That is, they provide analytical support to the decision-making process by offering recommendations based on data and structured models.

Among the various categories of DSS, Knowledge-Based Decision Support Systems (KB-DSS) stand out in this context. These systems incorporate repositories of specialized knowledge to generate recommendations tailored to the stakeholder's domain of application [4]. Additionally, KB-DSS are particularly valuable in scenarios where relevant decision-making knowledge is not easily quantified and requires the inference of qualitative insights [30].

In the domain of KT and software engineering, KB-DSS support the effective evaluation of KT by structuring both tacit and explicit knowledge into consultable processes organized by rules that guide future decisions [18]. Thus, these systems strategically assist in mitigating recurring practical limitations, such as the absence of clear criteria for KT evaluation, the difficulty in systematizing processes, and the informality of decision-making [21, 30].

3 Related Work

Although the evaluation of KT remains a relatively underexplored topic, existing studies in literature have proposed approaches to guide this process.

The study by [5] introduced an integrative conceptual model for evaluating KT in training contexts, categorizing its dimensions as summative (focused on outcomes), maintenance (long-term sustainability of results), and confirmative (continuous process improvement). However, the work remains at a conceptual level, without providing implementable artifacts, user interfaces, or data-driven decision-support mechanisms. In contrast, our solution integrates and operationalizes these conceptual dimensions within a tool, assisting stakeholders in conducting KT evaluation through the definition of objectives, the selection of processes, and decision-making support.

Inspired by the concept of organizational updating, [6] proposes a framework aimed at evaluating and improving KT processes in the academia–industry context. This framework classifies the level of formalization and evolution of KT processes; however, it does not provide operational or algorithmic resources to select or prioritize evaluation methods. In contrast, our approach offers a data-driven decision mechanism, without requiring a predefined level of organizational maturity. Furthermore, we implement a prioritization logic and a multi-layered architecture, enabling the delivery of an artifact that is both usable and applicable across diverse environments.

The study by [7] describes the development of a tool focused on the systematization of tacit KT, employing a fuzzy method for knowledge conversion. Although the proposal addresses knowledge conversion and systematization, it does not adopt process evaluation or comparison among available methods. In contrast, our solution fosters clarity in defining evaluation objectives, links the selected variables to the processes available in the systematized repository, and guides the practical execution of KT evaluation. This integration between systematization and evaluation, as presented in our tool, transcends existing solutions and proves to be both innovative and applicable to contexts that increasingly demand decision-support mechanisms.

4 Tool Development

The literature offers a broad range of methods, techniques, and tools intended to guide and support KT processes. However, the evaluation of these processes remains largely underexplored, with solutions often limited to conceptual models or support frameworks that lack empirical validation and practical adoption [13].

This study seeks to advance the understanding of the topic by providing an implemented solution that contributes to both KM and software engineering. Specifically, the objective is to develop a tool that assists organizations in evaluating their KT processes. This section details the tool's development process.

4.1 Initial Concept of the Solution

To understand how KT evaluation is conducted and to elicit the requirements necessary for the development of the proposed tool, both theoretical and practical investigations were carried out.

Initially, the document analysis technique was employed through an exploratory literature review [8], [9], [12]. This phase enabled the construction of an overview of the state of the art in KT evaluation, mapping existing models, theoretical gaps, and recurring patterns. Additionally, key elements of KT evaluation processes were identified and organized into a structured conceptual model, as presented in [10]. This conceptual model served as the foundation for the preliminary identification of the main functional requirements that a KT evaluation support tool should incorporate.

Based on the findings from the literature review, a more targeted and in-depth semi-structured interview protocol was developed to enhance the relevance of the empirical data collected and to understand how KT evaluation is conducted in practice. This protocol was applied in interviews with three professionals working in technology-based organizational environments, with 2 to 9 years of experience in evaluating KT processes. Therefore, the objective was not population exhaustiveness, but rather to achieve thematic saturation on the main topics of KT evaluation by exploring the qualified perceptions of these professionals [37]. After the interviews were conducted, they were transcribed and subjected to content analysis [15, 16], carried out by two researchers and supervised by two additional researchers.

The industry perspective made it possible to complement—and in some cases challenge—the theoretical findings, contributing to the validation, refinement, and expansion of the initially identified requirements by revealing aspects not addressed in the literature. Finally, the functional and non-functional requirements were presented to and validated by two of the interviewed professionals. It is important to note that due to scheduling constraints, validation from the third professional could not be obtained.

4.2 Tool Architecture

Based on the validated requirements, the application architecture was defined. In other words, the design patterns adopted, and the organizational structure of the information were established, ensuring the flow between the client interface (Angular), the layers (Controller, Service, Model), and the database MySQL (Figure 1).

4.2.1 Database Specification. The database construction was carried out in three successive stages, following the guidelines of [26]: conceptual modeling, logical modeling, and physical design. Initially, the conceptual modeling involved identifying entities based on the elicited functional requirements, defining their attributes, establishing relationships between these entities, and determining their cardinalities, resulting in the development of an entity-relationship diagram.

Subsequently, the conceptual model was converted into a logical model, in which entities were transformed into relational tables, applying integrity constraints and normalization, as well as verifying the consistency of the model. The physical design was then implemented using the MySQL database management system, selected for its strong performance for the expected use cases, broad compatibility with the adopted programming language, ease of integration with libraries and applications, developer-friendly interface, free licensing, and the wide availability of support materials and documentation.

4.2.2 Application Architecture Definition. The application specification was built based on a multi-layer client-server architecture, aiming to separate concerns, facilitate maintenance, and ensure scalability. The back-end, developed in Java using the Spring framework, was structured following the MSC (Model-Service-Controller) pattern, as it promotes code modularity and isolates business logic [31]. This approach contributed to clearer system organization and facilitated unit and integration testing.

Spring was chosen due to its maturity and support for modern web development practices, being widely adopted in the industry. The developed APIs follow RESTful principles, enabling communication with the front-end through standardized HTTP operations [34]. This strategy ensured application interoperability and facilitated future integration with other systems. The process prioritization logic used the Mergesort algorithm, whose O (n log n) complexity guarantees efficient performance even with large data volumes [3]. The decision to perform the sorting on the back-end aimed to optimize client memory and processing usage, reducing interface rendering time and improving the user experience in limited network environments [33].

The front end was implemented using the Angular framework due to its robustness in creating modular interfaces, its support for reactivity, its seamless integration with REST APIs, and its comprehensive documentation base, which contributed to development efficiency [32]. Interface design was supported by Figma, which enabled rapid prototyping [35]. For implementation, the ngx-admin template was adopted, providing a solid foundation of prebuilt components compatible with various libraries, accelerating development and ensuring visual consistency [34].

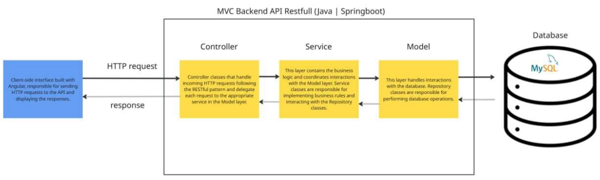

Figure 1: Layered Application Architecture

It is worth noting that the adopted architectural decisions allowed for a balance between performance, modularity, reusability, and testability. Moreover, the combination of Spring and Angular was selected because their architecture is considered extensible and suitable for agile development and maintenance [33]. This architectural cohesion also contributed to a shorter learning curve for new developers joining the project, streamlining onboarding and ensuring consistency in code contributions.

5 KnowEval

The KnowEval tool is a solution developed to support KT evaluation. In addition, it systematizes knowledge within its database, which is composed of these types of processes. KnowEval is operated by two main types of stakeholders: the end user and the administrator. Figure 2 provides an overview of the interaction between these stakeholders and the tool, illustrating the flow of knowledge systematization performed by the administrator and the user's journey through variable selection, process choice, and the conduction of KT evaluation based on the detailed process description.

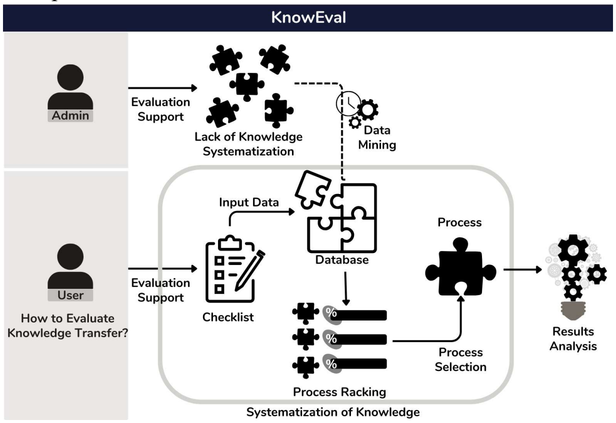

Figure 2: Overview of KnowEval

The perspectives of the specific roles and journeys of the end user and the administrator are detailed in Sections 5.2 and 5.3, respectively. It is important to emphasize that the tool does not automate KT evaluation; rather, it guides the user in selecting the process most appropriate for their context. Furthermore, the chosen process is composed of essential elements identified during the requirements elicitation phase.

5.1 Requirements Analysis

The structure and functionalities of the tool were based on requirements elicited through an exploratory literature review [10] and interviews with industry professionals. This requirements elicitation identified a set of key elements present in KT evaluation processes described in the literature. It was observed that most of these processes share common elements considered essential for structuring KnowEval, namely: well-defined stages and activity flows; a set of indicators and metrics; supporting tools for data collection and measurement; and the involvement of different stakeholder profiles.

In contrast, interviews[1] conducted in the industry revealed practical challenges. Professionals reported that the lack of systematized knowledge regarding KT evaluation has hindered its execution (as stated by Interviewee 2). Additionally, many KT evaluation processes found in the literature raise questions about their suitability to the specific application context. The interviewees described conducting KT evaluations in an ad hoc manner; facing difficulties in defining appropriate indicators and metrics; and identifying measurement failures due to insufficient measurable data. Although the industry lacks a formalized KT evaluation structure, we observed that it has a minimal structure somewhat similar to that identified in the literature.

Interviewee 2: *"The data found in the literature is all scattered; I find it difficult to evaluate a KT process ad hoc. I wish to find a repository with all the necessary information for the evaluation".*

In this context, we structured the KT evaluation processes obtained from the exploratory study based on the identified key elements. Subsequently, this dispersed knowledge was systematized and populated into the tool's database. During the structuring activity, we also extracted from the processes the objectives, characteristics, or strategies guiding the original processes, which we termed variables. These variables are represented in the end-user input checklist (Interviewee 3).

Interviewee 3: *"We define the objective, the characteristics, or the information we want from that evaluation. This is the essential starting point for the KT evaluation".*

Interviewee 1: *"We do not have a defined process for evaluating the transfer. We search the literature or partner organizations for processes that aim to evaluate what we need, and we use them".*

Additionally, we observed that processes addressing the greatest number of variables selected in the checklist tend to be more suitable for the end-user's application environment (according to Interviewee 1). Once located, this process is adopted. Thus, KnowEval (i) receives a set of variables as input data; (ii) systematizes the knowledge and searches the repository for KT evaluation processes, representing the processing via the database; and (iii) provides in the ranking a support mechanism for the end-user to choose a process, presented through graphical and textual representations detailing the key elements of the original process.

These requirements were validated without adjustment by the professionals experienced in KT evaluation (the interviewees), who confirmed that the tool satisfactorily met their needs. Subsequently, an entity-relationship model was constructed, defining the following entities: stakeholder (end-user), administrator (user with permission to manage the information available in the tool), process (set of information about the process), and alternative (input variables to locate a process). The resulting logical model was implemented in a MySQL database with version control managed via Flyway and connectivity to the back-end developed in SpringBoot. Finally, the main interfaces of the application were validated.

5.2 User Journey

KnowEval is a web-based tool designed to serve both specialist professionals and users with limited knowledge about KT evaluation. The user profiles encompass a wide range of organizational hierarchical levels, including directors, managers, analysts, technicians, among others, owing to its intuitive interface and the step-by-step guidance incorporated within the tool itself. Built from this perspective, the tool features a clear, intuitive, and

[1] https://abrir.link/IZGTN

user-centered interface aimed at ensuring ease of use and smooth navigation. The end user, who is responsible for conducting the KT evaluation, has access to three main interfaces: checklist, ranking, and processes.

In the checklist interface (Figure 3), the user is required to select options that represent the types of organizational data available or the kind of information they wish to consider in their evaluation context. The checklist is composed of strategic characteristics relevant to this type of evaluation, derived from processes identified in the literature, such as: knowledge nature (tacit, explicit, tacit and explicit, social, declarative, procedural); environment (inter-organizational or intra-organizational); organization type (public, private, research institution, startups); tools for data collection and measurement (questionnaires, interviews, workshops, evaluation indices, mathematical calculations); organization size (small, medium, large); type of artifact generated (report, theoretical model, recommendations, graphical representations); and timing of the evaluation execution (beginning, middle, or

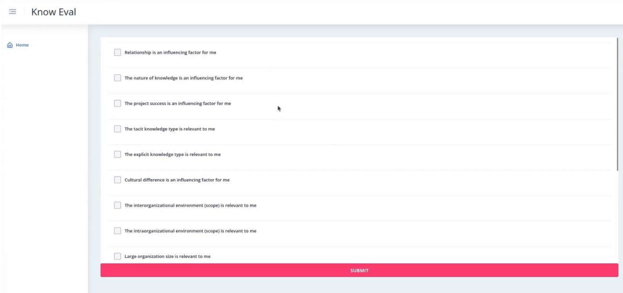

Figure 3: KnowEval Checklist Interface

After submitting their selection in the checklist, the user is redirected to the ranking interface (Figure 4), where the three processes that match the highest number of selected alternatives are displayed, along with their respective coverage percentages. Although a higher percentage generally suggests a more suitable process for the given context, the user retains full autonomy to select the process they deem most appropriate for conducting the KT evaluation.

Figure 4: Process Ranking Interface

In the process interface (Figure 5), the user gains detailed access to the process selected in the ranking. The tool provides a step-by-step guide to support the user throughout the KT evaluation, based on the key elements identified in the requirements elicitation. Accordingly, the document [2] structure includes a graphical representation of the selected process and a brief description of each element and its associated activities.

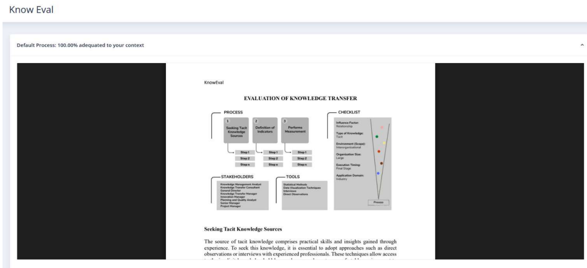

Figure 5: Process Document Description

By offering this level of support, the tool not only facilitates the evaluation process but also empowers users to make informed decisions based on the results obtained. This contributes to effective KM and fosters the continuous improvement of organizational processes.

5.3 Tool Management

The KnowEval administrator (KTadmin) plays a fundamental role in the operationalization of the tool, being responsible for the management and maintenance of the tool's database. Thus, the administrator has access to a menu with two main functions: management of the processes and management of the checklist variables. The main interface for this user is represented by a list of stored processes, displaying essential information such as identifier, name, and description (Figure 6). To ensure consistency and facilitate navigation and information retrieval, it is recommended that process names be standardized, and descriptions be clear, concise, and purpose-driven, aligned with the evaluative intent of each process.

The management of checklist alternatives is essential to the effectiveness of the tool in recommending evaluation processes that are aligned with the user's context. Additionally, the administrator contributes to structuring and registering the descriptive elements of each process. KnowEval architecture supports dynamic expansion of the database, allowing the insertion of new checklist characteristics or their categorization as the complexity and diversity of evaluation contexts become more apparent.

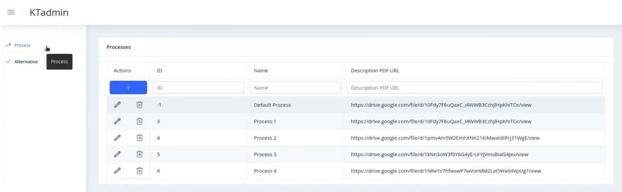

Figure 6: Process Management Interface

The management of KnowEval directly contributes to its flexibility and scalability, enabling the continuous evolution of its process repository. Moreover, by promoting centralized data management, it is possible to ensure the integrity and currency of the tool, as well as to guarantee that the recommended processes remain relevant and contextually appropriate.

[2] Demonstration of the Process Detailing: https://abrir.link/dEkOb

6 Evaluation

The evaluation of KnowEval was designed to verify its effectiveness in supporting the assessment of KT within organizational contexts, with particular emphasis on software development environments. To this end, a qualitative analysis was carried out based on feedback collected from the interviewees who contributed to the requirements elicitation process, namely two software developers and one researcher with expertise in KT and empirical software engineering.

The evaluation of the tool focused on two criteria: (i) the utility of the tool in supporting decision-making related to KT evaluation processes, and (ii) the usability of the interface for both categories of users. Accordingly, the utility criterion sought to determine whether the features offered by KnowEval effectively contribute to reducing subjectivity and adding value to the KT evaluation activities performed by users. The usability criterion, in turn, was employed to assess the ease of navigation, with the objective of ensuring that KnowEval does not require excessive learning effort and can be used by individuals with limited prior experience in this type of tool.

For this purpose, each participant was individually presented with a hypothetical scenario that allowed them to navigate through the stages of the tool designed to support the evaluation of KT. Initially, the participants accessed the checklist screen, that is, they randomly selected input variables, namely: organization size, influencing factor, generated artifact, context, indicator, and measurement tool. Subsequently, they navigated through the process-ranking screen, where they selected a KT evaluation process of their choice, and then proceeded to review the detailed description of the process in the document. The participants were invited to record, under each criterion, their perceptions regarding the checklist, the process ranking, and the process detailing. Figure 7 presents the variables selected by each participant in the hypothetical scenario, along with some of the participants' perceptions of KnowEval.

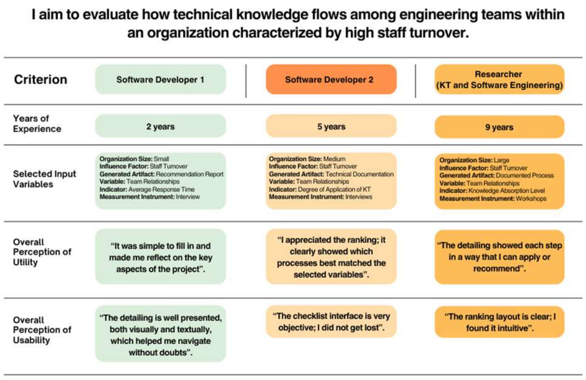

Figure 7: Overview of the KnowEval Evaluation

In terms of utility, the developers emphasized the tool's applicability within software companies, highlighting its relevance in projects with high developer turnover; in contexts involving project transitions, where KT actions must be systematized; and in scenarios that require justification of KT, such as onboarding new team members. The researcher, in turn, underscored the formalization of practices enabled using KnowEval, as well as the variable-process matching mechanism, which has the potential to reduce the subjectivity in selecting KT evaluation methods.

Regarding usability, all evaluators praised the intuitive interface, the clear integration flow, and the accessible language. The smooth navigation between the main screens of the tool was found to be understandable even for a developer with limited professional experience. Finally, all evaluators considered KnowEval to be an innovative and promising contribution to software engineering, as it succeeds in preliminarily systematizing KT knowledge, thereby reducing the effort and rework for professionals who are required to conduct KT evaluations in addition to their core responsibilities.

7 Threats to Validity and Limitations

Despite the positive results obtained from the evaluation of KnowEval, it is important to acknowledge certain limitations, namely: (i) the quantitatively limited sample size, which allowed for the extraction of relevant insights but lacks broader evaluative scope across real-world scenarios; (ii) the evaluation criteria, which need to be expanded to yield more robust insights into the tool's overall quality; and (iii) the KnowEval database, which, although sufficient for initial testing, requires expansion to enhance the systematization of knowledge.

8 Final Considerations

KnowEval seeks to minimize the challenges faced by stakeholders in the post-mortem analysis of KT processes. In this regard, the tool not only supports KT evaluation but also systematizes knowledge on the subject, which is currently scattered throughout the literature; provides decision-support mechanisms that optimize user activities; transcends purely conceptual solutions through its operational applicability; and establishes acceptable parameters for practical adoption in real-world environments, as evidenced by its promising validation.

In the future, KnowEval will undergo further refinement to address its current limitations, including the development of a user authentication environment to enable response history tracking, as well as the construction of a geographic support network aimed at generating insights into the collaborative practice of KT evaluation.

ARTIFACT AVAILABILITY

The supplementary material is available through the user interface [38], the administrator interface [39], and the API [40]. The tool is released under the MIT License [36].

ACKNOWLEDGMENTS

This study was partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brazil (CAPES), Finance Code 001. This work is partially supported by INES.IA(www.ines.org.br), CNPq grant 408817/2024-0. Sérgio Soares is partially supported by CNPq grant 306000/2022-9.

REFERENCES

[1] R. V. D. Gonzalez and M. F. Martins. 2017. Knowledge Management Process: Theoretical conceptual research. *Gestão & Produção* 24, 1 (2017), 248–265.

[2] Z. Gu, F. Meng, and M. Farrukh. 2021. Mapping the Research on Knowledge Transfer: A Scientometrics Approach. IEEE Access 9 (2021), 34647–34659.

[3] Y. Luo and Y. Zhou. 2016. A Decision Support System for Supplier Selection Using an Improved Mergesort Algorithm. Mathematical Problems in Engineering 2016 (2016), Article 1–13.

[4] Cinelli, M., Kadziński, M., Miebs, G., Gonzalez, M., & Słowiński, R. 2021. Recommending multiple criteria decision analysis methods with a new taxonomy-based decision support system.

[5] Perez-Soltero, A., Aguilar-Bernal, C., Barceló-Valenzuela, M., Sánchez-Schmitz, G., Meroño-Cerdán, A. L., & Fornés-Rivera, R. (2019). *Knowledge Transfer in Training Processes: Towards an Integrative Evaluation Model. IUP Journal of Knowledge Management*, 17(1), 7–40.

[6] Scanlan, J. (2018). *A Capability Maturity Framework for Knowledge Transfer. Industry and Higher Education*, 32(4), 235–244.

[7] Patalas-Maliszewska, J., & Krebs, I. (2016). *A Model of the Tacit Knowledge Transfer Support Tool: CKnow-Board*. In G. Dregvaite & R. Damasevicius (Eds.), *Information and Software Technologies* (pp. 30–41). Springer, Cham.

[8] Campbell, A., Carvill, E., Elliott, R.: Annual Review and Annual Knowledge Transfer Sur-vey, 2018. Knowledge Transfer Ireland - KTI, Where Research & Business Connect (2019).

[9] Tian, H., Guan, C.: Evaluation of the Effectiveness of Knowledge Transfer between Enter-prise Mentoring and Apprenticeship Based on the AHP-FCE Model Set. International Con-ference on Big Data Economy and Information Management (2023).

[10] W. Ferreira, J. Saraiva, and S. Soares. " A Conceptual Framework to Support Organizational Knowledge Transfer Evaluation", in Anais Estendidos do XXI Simpósio Brasileiro de Sistemas de Informação, Recife/PE, 2025, pp. 345-351, doi: https://doi.org/10.5753/sbsi_estendido.2025.246898.

[11] Claudia P. C. Maciel, E. F. S., Ricardo de A. F., Katia R. F., and Nandamudi L. V. 2018. Knowledge Management Diagnostics in Software Development Organizations: a Systematic Literature Review. In Proceedings of the XVII Brazilian Symposium on Software Quality (SBQS '18). Association for Computing Machinery, New York, NY, USA, 141–150. https://doi.org/10.1145/3275245.3275260

[12] M. Rohrbach, M. Stark and B. Schiele, "Evaluating knowledge transfer and zero-shot learning in a large-scale setting," CVPR 2011, Colorado Springs, CO, USA, 2011, pp. 1641-1648, doi: 10.1109/CVPR.2011.5995627.

[13] Cunha, F. O. M., Perkusich, M., Almeida, H. O., Gorgônio, K., & Perkusich, A. (2022). TeamPlus: A Decision Support System for Software Team Formation. In Proceedings of the 36th Brazilian Symposium on Software Engineering (SBES 2022) (pp. 118–123). Association for Computing Machinery. https://doi.org/10.1145/3555228.3555275.

[14] Milagres, R., Burcharth, A. 2018. Knowledge transfer in interorganizational partnerships: what do we know? Business Process Management Journal, 25, 27-68.

[15] Bardin, L.: Content Analysis. São Paulo: Ed. 70 (2011).

[16] Lima, J. L.O., Manini, M. P.: Methodology for Qualitative Content Analysis with the technique of Mind Maps using Nvivo and FreeMind softwares. Inf. Inf., L, 21, 63 – 100 (2016).

[17] Dellermann, D., Lipusch, N., Ebel, P., & Leimeister, J. M. (2021). Design principles for a hybrid intelligence decision support system for business model validation.

[18] Almalki, S. S. (2025). AI-Driven Decision Support Systems in Agile Software Project Management: Enhancing Risk Mitigation and Resource Allocation. Systems, 13(3), 208. https://doi.org/10.3390/systems13030208.

[19] Saray Galeano Ospino, Liliana Machuca-Villegas, Gloria Piedad Gasca-Hurtado. 2023. Knowledge Transfer in Software Development Teams: a Gamification Strategy to Encourage Collaboration. Revista Ingenierias Universidad de Medellín. https://doi.org/10.22395/rium.v21n41a2

[20] Lihong Zhang, Saeed Banihashemi, Liting Zhu, Homa Molavi, Eyyub Odacioglu, Miyuan Shan. A scientometric analysis of knowledge transfer partnerships in digital transformation. Journal of Open Innovation: Technology, Market, and Complexity, Volume 10, Issue 3, 2024, 100325, ISSN 2199-8531, https://doi.org/10.1016/j.joitmc.2024.100325.

[21] S. S. Almalki. 2025. AI-Driven Decision Support Systems in Agile Software Project Management: Enhancing Risk Mitigation and Resource Allocation. *Systems*, 13(3), 208. DOI: https://doi.org/10.3390/systems13030208

[22] D. Marijan and A. Gotlieb. 2022. Industry-Academia Research Collaboration in Software Engineering: The Certus Model. *arXiv preprint arXiv:2204.11039*. DOI: https://doi.org/10.48550/arXiv.2204.11039

[23] B. Cartaxo, G. Pinto, and S. Soares. 2020. Rapid Reviews in Software Engineering. arXiv preprint arXiv:2003.10006. DOI: https://doi.org/10.48550/arXiv.2003.10006

[24] F. O. Bjørnson and T. Dingsøyr. 2018. Knowledge Management in Software Engineering: A Systematic Review of Studied Concepts, Findings and Research Methods Used. arXiv preprint arXiv:1811.12278. DOI: https://doi.org/10.48550/arXiv.1811.12278

[25] Fredrik Sunnemark, Wilma Lundqvist Westin, Tamy Al Saad, Per Assmo. 2024. Exploring barriers and facilitators to knowledge transfer and learning processes through a cross-departmental collaborative project in a municipal organization, The Learning Organization, Volume 31, Issue 3, 2024, Pages 358-374, ISSN 0969-6474, https://doi.org/10.1108/TLO-01-2023-0003.

[26] Silberschatz, A., Korth, H. F., & Sudarshan, S. 2010. Database System Concepts. McGraw-Hill.

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

[28] Smith, E.A. 2001, "The role of tacit and explicit knowledge in the workplace", Journal of Knowledge Management, Vol. 5 No. 4, pp. 311-321. https://doi.org/10.1108/13673270110411733

[29] Qian, X., Liu, X., Pu, R. 2020. Evaluation of Knowledge Transfer Efficiency Based on the Synergy of Innovative Clusters. Journal of Creative Entrepreneurship and Management, Vol. 1, No. 2, pp. 1-25.

[30] Hamrouni, B.; Bourouis, A.; Korichi, A.; Brahmi, M. 2021. Explainable Ontology-Based Intelligent Decision Support System for Business Model Design and Sustainability. *Sustainability*, *13*, 9819. https://doi.org/10.3390/su13179819.

[31] Christian Ullenboom. 2023. Spring Boot 3 and Spring Framework 6. Rheinwerk Computing, Bonn, Germany.

[32] Diptiben Ghelani, Tan Kian Hua, Surendra Kumar Reddy Koduru. (2022). A Model-Driven Approach for Online Banking Application Using AngularJS Framework. American Journal of Information Science and Technology, 6(3), 52-63. https://doi.org/10.11648/j.ajist.20220603.12

[33] Bass, Len, Paul Clements, and Rick Kazman.2021. *Software architecture in practice*. Addison-Wesley Professional.

[34] Madurapperuma, I. H., M. S. Shafana, and M. J. A. Sabani. 2022. State-of-art frameworks for front-end and back-end web development.

[35] Al-Faruq, M. N. M., Nur'aini, S., & Aufan, M. H. (2022). Perancangan Ui/Ux Semarang Virtual Tourism Dengan Figma. Walisongo Journal of Information Technology, 4(1), 43-52.

[36] Augusto Campos. 2006. O que é software livre. Disponível em: http://br-linux.org/linux/faq-softwarelivre. Acesso em: 20 de maio de 2025.

[37] Guest, G.; Namey, E.; Mckenna, K. How many focus groups are enough? Building an evidence base for nonprobability sample sizes. Field Methods, v. 29, n. 1, p. 3–22, 2017. https://doi.org/10.1177/1525822X16639015.

[38] https://zenodo.org/records/15485171

[39] https://zenodo.org/records/15485199

[40] https://zenodo.org/records/15485285