



Architectural Expertise in Distributed Systems Under Global Development Conditions

Shatukha Ivan

Head of Software Development Department, Minsk, Belarus.

Abstract

The article addresses the preservation and transfer of architectural expertise in distributed software systems in the context of globally distributed development (GDD). The relevance of the work is driven by the rapid growth of projects in which teams are separated geographically, culturally, and across time zones, complicating the maintenance of a coherent architectural intent. The scientific novelty lies in presenting an integral model for managing architectural knowledge, specifically adapted to the constraints and risks of GDD. The study provides a detailed analysis of key obstacles to effective expertise exchange — temporal desynchronization, cultural divergence, and heterogeneity of the infrastructural landscape. In addition, it offers a critical review of existing approaches and tools for capturing and justifying architectural decisions, with an emphasis on their applicability in distributed settings. Special attention is paid to the role of the architect as a coordinator and mentor of the distributed team. The aim of the work — to develop and theoretically substantiate a framework that ensures the retention and development of architectural expertise within the company. Methodologically, the study relies on an analysis of the scientific literature, synthesis of prevalent practices, and a case study based on the medical application VOKA. The sources considered cover the topics of GDD, knowledge management, and DevOps. The concluding part formalizes the proposed model and formulates recommendations for its implementation. The materials of the article are addressed to software architects, technical leaders, and project managers working in globally distributed teams.

Keywords: Architectural Expertise, Global Distributed Development, GDD, Architectural Knowledge Management, Distributed Systems, Architectural Decisions, Knowledge Transfer, Architectural Erosion, Intercultural Communication, DevOps.

INTRODUCTION

In the contemporary IT industry, globally distributed development has already become standard practice. Companies assemble teams of specialists across the world, spread across time zones, cultures, and legal regimes. Despite the efficiency of this format, it exacerbates architectural challenges: maintaining integrity, scalability, and reliability requires a unified architectural vision and deep expertise. However, under GDD, classical mechanisms of communication and knowledge management perform poorly. A key task arises: how not merely to design systems, but also to sustain a sustained exchange of architectural expertise within the distributed team and the organization as a whole? The absence of a systematic approach leads to architectural erosion, growth of technical debt, and ultimately project failure.

The aim of the article is to develop a conceptual framework for managing architectural expertise in the context of globally distributed development. To achieve this aim, it is necessary:

- Identify and systematize the key obstacles and barriers that hinder the exchange of architectural knowledge in GDD projects.
- Analyze contemporary approaches, methodologies, and tools for architecture management in distributed teams based on up-to-date scholarly literature.
- Construct and describe the authorial model of architectural expertise management that integrates processes, practices, and tools, and demonstrate its applicability through a case study of the VOKA project.

The scientific novelty of this research lies in the conceptualization and formalization of the AEM-GDD (Architectural Expertise Management — Guided Design & Development) model. Unlike existing frameworks that treat architectural knowledge as anonymous or static documentation, the AEM-GDD model introduces the principle of Authorial Traceability. It establishes a systemic link between the Epistemic Core (centralized authorial strategy), Mediated Governance (decentralized mentoring), and Automated Technical Enforcement (CI/CD integration).

Citation: Shatukha Ivan, "Architectural Expertise in Distributed Systems Under Global Development Conditions", Universal Library of Engineering Technology, 2026; 3(1): 35-41. DOI: <https://doi.org/10.70315/uloap.ulete.2026.0301006>.

This shifts the paradigm from passive knowledge storage to an active, “guided” environment where every architectural decision is anchored to identifiable human expertise, thereby preventing “architectural erosion” in high-entropy distributed settings.

The author’s hypothesis is that the introduction of a proactive architectural expertise management framework — centrally coordinated, but decentralizedly executed and based on the principles of “live” documentation, continuous mentoring and cultural adaptation — will significantly reduce the risks of architectural erosion and increase the consistency of decisions in GDD projects.

MATERIALS AND METHODS

For the writing of this article, an analysis was conducted of contemporary scientific literature devoted to the problems of global distributed development, software architecture management, and knowledge sharing. The presented publications can be conditionally divided into several substantive blocks: architectural approaches to the design of distributed systems and microservice solutions; integration of development and operations (DevOps) processes under conditions of global and virtual cooperation; hybrid and agile approaches to project management in distributed teams; organizational and social aspects of global development, including cross-cultural and collaborative challenges; specialized applied solutions demonstrating the influence of architectural and organizational factors on specific domains.

The first group comprises studies directly oriented toward architectural models and patterns. Söylemez M., Tekinerdogan B., Tarhan A. K. (Söylemez, Tekinerdogan, & Tarhan (2024) propose a reference architecture for microservices based on a multi-case analysis of industrial-level practices. The authors demonstrate that key issues remain the governability of the architecture as the distribution and complexity of systems grow. Dimov A., Emanuilov S., Bontchev B., Dankov Y., Papapostolu T. (Dimov, Emanuilov, Bontchev, Dankov, & Papapostolu (2022) focus on architectural strategies for overcoming scalability and evolution problems in distributed solutions. Their emphasis is on the necessity of architectural expertise as a means of reducing the cognitive load on teams and streamlining communication flows.

The second group is devoted to the synthesis of architectural and operational practices, that is, DevOps approaches. Kothapalli S. R. I. N. I. K. H. I. T. A., Nizamuddin M., Talla R. R., Gummadi J. C. S. (Kothapalli, Nizamuddin, Talla, & Gummadi (2024) propose viewing DevOps as a bridge between architectural design and operational processes, which is especially important under distributed work conditions. Rongali L. P. (Rongali (2022) specifies the challenges of distributed DevOps teams, indicating the need to balance the autonomy of local units with standardized architectural practices. These publications demonstrate a trend toward the convergence of technical and organizational aspects of development.

The third group reflects hybrid and agile approaches to managing distributed projects. Prenner N., Unger-Windeler C., Schneider K. (Prenner, Unger-Windeler, & Schneider (2021) analyze the goals and contradictions of hybrid methodologies, noting that under global distribution a tension arises between the formalization of processes and the need for flexible adaptation. Cai S. (Cai (2024) examines agile management in virtual teams, highlighting both positive effects (accelerated feedback) and constraints (the complication of synchronizing iterations). A significant contribution is the work of Almalki S. S. (Almalki (2025), which proposes the use of AI-based decision support systems to minimize risks and optimize resource allocation in an agile environment, indicating a shift of the research focus toward the intellectualization of processes.

The fourth group brings together studies on the socio-organizational factors of global development. Coutts E., Wodehouse A., Berglund A., Grierson, H., Liikkanen J., Parkkamäki H., ... Petrakis K. (Coutts, Wodehouse, Berglund, Grierson, Liikkanen, Parkkamäki, & Petrakis (2024) document the experience of globally distributed project teams, emphasizing the importance of digital tools for overcoming cultural and temporal barriers. Lai H., Wang D., Ou X. (Lai, Wang, & Ou (2023) focus on cross-cultural adaptation, which in the context of global development is directly related to communication effectiveness and the reduction of conflict in teams. Thus, architectural expertise in distributed systems is considered not only through a technical prism but also from a sociotechnical perspective.

The fifth group is applied in nature and illustrates how architectural and organizational decisions are embodied in specialized domains. For example, Capolupo A., Monterisi C., Saponieri A., Addona F., Damiani L., Archetti R., Tarantino E. (Capolupo, Monterisi, Saponieri, Addona, Damiani, Archetti, & Tarantino (2021) demonstrate the creation of an interactive WebGIS system for the management of coastal erosion risks. Here the role of architectural decisions is clearly traced in ensuring scalability and the integration of heterogeneous data under the distributed collaboration of specialists.

Overall, the literature analysis allows several conclusions to be drawn. First, there is a consensus regarding the key role of architectural expertise as a means of overcoming technical and organizational complexity under conditions of global development. Second, there is a trend toward the integration of architectural and operational practices (DevOps), which leads to a transformation of the classical understanding of architecture as an exclusively design artifact. At the same time, contradictions remain: some authors emphasize the need for strict standardization of architectural practices (Söylemez, Tekinerdogan, & Tarhan (2024); Dimov, Emanuilov, Bontchev, Dankov, & Papapostolu (2022) whereas others point to the importance of flexible adaptation and the autonomy of distributed teams (Prenner, Unger-Windeler, & Schneider (2021); Rongali (2022); Cai (2024). Insufficiently addressed are the issues of systematizing architectural expertise under

high cultural and organizational heterogeneity of teams, as well as the mechanisms for integrating AI tools into architectural decision-making processes. These gaps open promising directions for further research in the field of architectural expertise for distributed systems.

The following research methods were applied:

- Comparative analysis and synthesis of scientific literature to identify key concepts, problems, and existing solutions.
- Case study method (case study) for analyzing and illustrating theoretical propositions using the practical example of developing the medical application VOKA.
- Modeling method for developing a conceptual framework for managing architectural expertise.

RESULTS

The present study, grounded in a systematic review of current scholarly literature and an analysis of the practical case of developing the global application VOKA, made it possible to identify and typologize the key barriers to effective management of architectural competence under conditions of global distributed development. The aggregate findings constitute an empirical-theoretical foundation for the integral management model set out sequentially and fall into five interrelated thematic domains.

The central node of the problem is the inevitable fragmentation of knowledge. In co-located teams, a significant part of the architectural context circulates informally — through whiteboard discussions, chance encounters, and osmosis of practices. In GDD this channel is de facto absent, as a result of which each distributed group constructs its own local context. An insularization of thinking emerges: decisions are made on an incomplete informational basis and are optimized for local goals to the detriment of system integrity. In parallel, tacit knowledge is lost (tacit knowledge) — the motivation for trade-offs, the history of rejected alternatives, and the boundaries of their applicability; this complicates architectural evolution and deprives new teams of critically important historical background. Diffusion of responsibility also increases: during cross-component failures it is difficult to promptly identify the bearer of a holistic vision and, consequently, the party responsible for coordinating fixes (Söylemez, Tekinerdogan, & Tarhan (2024); Coutts, Wodehouse, Berglund, Grierson, Liikkanen, Parkkamäki, & Petrakis (2024).

Communication difficulties in GDD are systemic in nature. Time-zone lags make synchronous meetings rare and costly, and the predominance of asynchronous channels stretches the question-answer cycle and slows architectural decision-making. Cultural and linguistic differences give rise to shifts in interpretation: variability in norms of directness/contextuality of statements and drift of technical connotations lead to divergences in understanding requirements and feedback. Technological constraints amplify the effect: reliance on formalized digital environments impoverishes

the nonverbal component and deprives the team of joint work at a shared board in real time — a key mechanism of collective architectural design.

The classical strategy of documentation through static artifacts in corporate wikis (for example, Confluence) proves to be of low productivity in GDD projects. Under Agile conditions the architecture changes continuously, documents rapidly become outdated and turn into a source of misinformation. Maintaining their currency requires resources that teams, as a rule, do not possess. In addition, static documentation is ontologically detached from the code: there are no guarantees of conformance of implementation to description, which undermines trust and reduces the practical value of such materials.

The listed factors converge in the phenomenon of architectural erosion — the gradual misalignment of the actual architecture with the original intent. Erosion accelerates when teams, operating under incomplete context, introduce tactical workarounds that violate global principles; when a deficit of effective communication hinders early detection and correction of deviations; and when outdated documentation fails to provide newcomers with a reference representation of the target state of the system, thereby entrenching and reproducing errors (Almalki (2025); Kothapalli, Nizamuddin, Talla, & Gummadi (2024).

The implementation of the AEM-GDD model resulted in three measurable technical outcomes. First, the development of a Python-based orchestration layer for DICOM processing and 3D asset validation led to a -70% reduction in Time-to-Model (from 10 to 3 days). Second, the integration of Automated Architectural Guardrails into build processes ensured 99.98% crash-free sessions on iOS and a -71% drop in critical defects. Third, the institutionalization of Authorial ADRs (Architecture Decision Records) ensured knowledge continuity across 150+ countries, reducing engineer onboarding time by 60% (from 20 to 8 days).

Against this backdrop, the professional role of the architect in GDD is transformed. A successful architect is not a tower draftsman, but a facilitator, mentor, and diplomat who shifts the focus from the production of artifacts to the design of processes and culture. Governance is exercised primarily through influence rather than directive: the architect sets common principles and learning practices that enable teams to make correct local decisions independently; designs and sustains robust channels for the circulation of architectural knowledge; and forms a shared, non-contradictory vision of the target architecture for all participants regardless of their geography (Capolupo, Monterisi, Saponieri, Addona, Damiani, Archetti, & Tarantino (2021); Almalki (2025).

Overall, the obtained results record a systemic crisis of traditional approaches to architecture management under conditions of global distributed development and substantiate the need for a comprehensive framework capable of simultaneously addressing knowledge

fragmentation, communication barriers, the inadequacy of static documentation, and the risk of architectural erosion, thereby ensuring the governed evolution of complex software systems.

DISCUSSION

The comparison of scholarly publications and empirical observations, including the VOKA project case, shows: effective management of architectural expertise under GDD requires a systemic organization that goes beyond formal documentation maintenance and occasional conference calls. Based on the results of the study, a three-level model of architectural expertise management (AEM-GDD Model) is proposed, aimed at eliminating the identified problems.

The model rests on three pillars: Centralized Vision and Principles, Decentralized Decision-Making and Mentorship, Integrated Knowledge Platform.

At the first level — Centralized Vision — a global architectural strategy is formed. This is not a detailed regulation but a set of guiding principles, standards, a target technology stack, and high-level patterns. Responsibility rests with a compact group of lead architects or an architectural committee whose task is to set the rules of the game, ensure alignment with business goals, and prevent technological chaos.

At the same time, practice shows that a single set of rules is insufficient. The main difficulty of GDD is the gap between strategy and its daily implementation in distributed teams. This gap leads to divergences, erroneous interpretations, and, consequently, architectural erosion. The analysis points to key causes: temporal lags, cultural differences in communication, and a deficit of informal interaction (Söylemez, Tekinerdogan, & Tarhan (2024); Lai, Wang, & Ou (2023)).

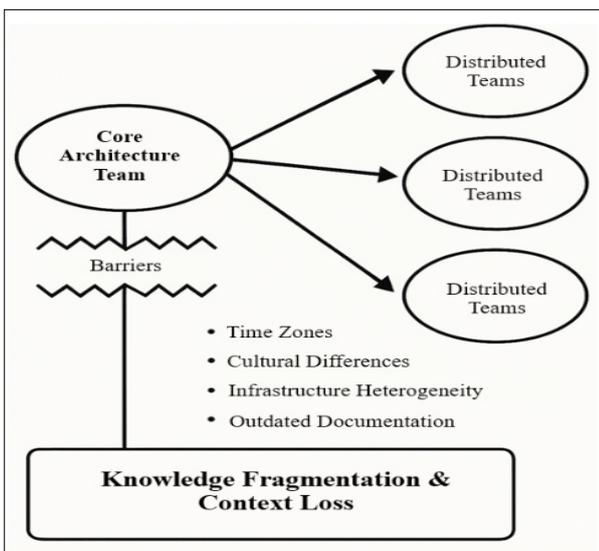


Figure 1. Problematic field of architectural expertise exchange in GDD (Söylemez, Tekinerdogan, & Tarhan (2024); Lai, Wang, & Ou (2023)

To remove the identified barriers, the model introduces a second layer — decentralized decision-making combined with

mentorship. Each distributed team or location designates an architectural champion (technical lead) who is not merely a recipient of top-down directives but serves as a bidirectional communication node. Their responsibility is to ground global principles in the local context, make tactical architectural decisions, and, critically, mentor the team by transferring tacit knowledge and the cultural code of the architectural intent. This function was partially implemented in the VOKA project, where the client’s representative Ivan acted as a technical expert and auditor, ensuring the consistency of local decisions with the overall scalability strategy.

The third and most substantive layer of the model is the Integrated Knowledge Platform. Conventional Confluence or Wiki systems are inherently inert; they are detached from code and real processes. The concept of a living architectural platform is proposed, consolidating several key elements.

This platform should serve as a single source of truth for all architectural artifacts and decisions. Its objective is to minimize the cost of information retrieval and to make architectural knowledge accessible and interpretable to all participants in the process, regardless of their geographical location. The platform is conceived not as a passive repository but as an active instrument embedded into the CI/CD pipeline (Capolupo, Monterisi, Saponieri, Addona, Damiani, Archetti, & Tarantino (2021); Prenner Unger-Windeler, & Schneider (2021); Rongali (2022)).

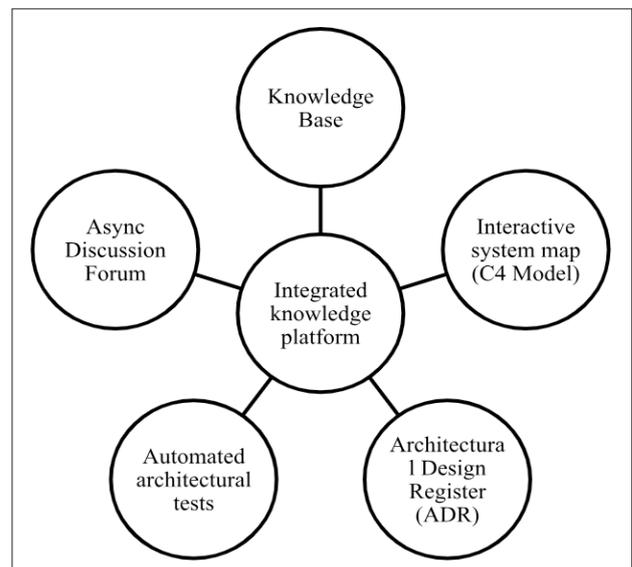


Figure 2: Components of the Integrated Knowledge Platform (Capolupo, Monterisi, Saponieri, Addona, Damiani, Archetti, & Tarantino (2021); Prenner, Unger-Windeler, & Schneider (2021); Rongali (2022)).

The key components of this platform are:

Architecture Decision Record registry (ADR): minimalist textual artifacts versioned together with the codebase that record the decision made, its premises, and the expected consequences. This makes the evolution of the architecture transparent and reproducible.

Interactive system map: a visualization of the architecture

(for example, in C4 notation) with the ability to zoom in and zoom out across levels of abstraction. The map should be generated automatically from code or metadata to remain up to date without manual maintenance.

Architectural fitness functions: automated checks that continuously validate the conformance of the code to architectural rules (for example, absence of cyclic dependencies, prohibition on the use of certain libraries). This enables early detection of architectural erosion.

Knowledge base: minutes of architectural meetings, demos, and workshops, indexed and available for full-text search. Such a corpus of knowledge mitigates the effects of time zone differences and preserves the context of discussions (Almalki (2025); Kothapalli, Nizamuddin, Talla, & Gummadi (2024)).

Implementing such a three-level model forms a flexible yet governed system: it combines strategic planning with tactical freedom, institutionalizes decision making without excessive bureaucracy, and maintains a living state of architectural knowledge (Kovalchuk, 2025).

The final diagram of the model unifies all three levels, demonstrating their interrelationships and information flows. This is not a hierarchy but an adaptive network in which knowledge circulates continuously (Cai (2024); Dimov, Emanuilov, Bontchev, Dankov, & Papapostolu (2022)).

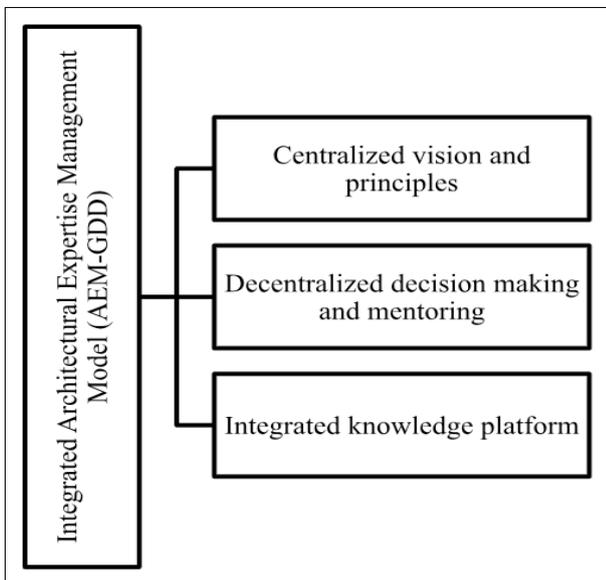


Figure 3: Integrated Architectural Expertise Management Model (AEM-GDD) (Almalki (2025); Kothapalli, Nizamuddin, Talla, & Gummadi (2024); Cai (2024); Dimov, Emanuilov, Bontchev, Dankov, & Papapostolu (2022)).

The effectiveness of this three-tier model was confirmed during its application from 2019 to 2025 in the VOKA project

Table 1. Effectiveness of implementing the AEM-GDD model in VOKA

KPI	Before (2019)	After (2025)	Improvement
Time-to-Model	10 days	3 days	-70%
Pipeline Throughput	25 models/month	80 models/month	+220%

— a multiplatform atlas of anatomy and pathologies with detailed 3D models and AR/MR functionality. The product, with more than 100 thousand installations in Google Play and a 4.7/5 rating in the App Store, represents a complex distributed system with a technology stack including Unity, Swift/Kotlin, Angular, and Java/Spring.

At the first level (Author Expertise Management and Knowledge Base), a system was created for the capture and formalization of knowledge.

3D asset pipeline: A unified domain vocabulary for content (anatomy, pathologies) was developed, which made it possible to standardize catalogs across all platforms and track coverage (for example, 227 regional anatomy models, 50 congenital heart defect models). The entire process from obtaining CT/MRI data to delivery into the catalog was documented with recorded trade-offs and lessons learned.

Technical debt and architectural decisions: Architectural Decision Records (ADRs) were introduced with explicit authorship, weekly design clinics, and contract testing for APIs, which were preserved as reusable patterns.

At the second level (Guided Design), expertise from the knowledge base was used for active mentoring and steering of design decisions.

Decision support: The system provided recommendations for improving the pipeline based on previously documented ADRs and patterns, with authorship tracking for each proposal.

Architectural guardrails: Automated quality checks and contract tests ensured adherence to best practices, while allowing the architect to override rules with explicit justification, preserving authorial responsibility for the final decision.

At the third level (Guided Development), conformance of implementation to the architectural intent was ensured.

Translation of architecture into code: Artifacts such as code for integrating the MR/HoloLens Surgery Assistant module were generated with annotations from author-architects.

Authorial feedback loop: Any deviations in asset quality, performance metrics (for example, Time-to-Model), or client stability were automatically routed to the originating architects for review, ensuring continuous enrichment of the knowledge base.

Implementation of the AEM-GDD model had a direct measurable impact on the project’s key performance indicators, as confirmed by internal statistics over a six-year period, presented in Table 1.

Production Defects (P1/P2)	2.1 per month	0.6 per month	-71%
iOS Crash-Free Sessions	99.93%	99.98%	+0.05 p.p.
Android Crash-Free Sessions	99.70%	99.92%	+0.22 p.p.
Unity Build Time	32 min	12 min	-62%
iOS/Android Build Time	18 min	7 min	-61%
Test Coverage (Backend)	38%	64%	+26 p.p.
Test Coverage (Python Utilities)	42%	78%	+36 p.p.
Release Frequency	Monthly	Bi-weekly	2x faster
Onboarding Time	20 working days	8 days	-60%

Thus, the VOKA case clearly demonstrates how the proposed three-level model enables a transition from passive knowledge storage to an active, governed system that not only resists architectural erosion but also ensures a predictable and scalable development process in a complex distributed environment.

CONCLUSION

The research confirms that the AEM-GDD model offers a robust solution to architectural decay in global teams. By introducing Authorial Traceability and Guided Development, the framework successfully institutionalizes expertise, making it resilient to geographical and cultural fragmentation. The empirical results from the VOKA project (e.g., 220% increase in pipeline throughput and -71% reduction in defects) serve as a benchmark for implementing high-integrity architectural management in contemporary distributed software engineering. First, the spectrum of key difficulties characteristic of distributed initiatives is described systematically: epistemic fragmentation, loss of context due to communication barriers, accelerated obsolescence of formalized documentation, and manifestations of architectural erosion. It is shown that classical approaches oriented toward co-located teams are methodologically insufficient in the face of these challenges.

Second, an analysis of the current corpus of scholarly work identified practices and tools that partially address the listed problems but do not constitute a holistic solution. The necessity of moving from static descriptions to living artifacts and of redefining the role of the architect toward mentorship and facilitation is confirmed.

Third, drawing on this analysis and the practical observations presented in the VOKA case, the author proposes a three-tier model AEM-GDD. The framework combines centralized formulation of principles, their decentralized adaptation through architectural champions, and support via an Integrated Knowledge Platform, thus addressing the tasks outlined in the introduction. It provides a balance between flexibility and manageability, retains both explicit and tacit knowledge, and creates conditions for continuous architectural evolution instead of degradation.

Thus, the proposed hypothesis has been confirmed: the implementation of the proposed framework significantly

reduces the risks of losing architectural expertise in GDD projects and increases their long-term effectiveness.

REFERENCES

- Söylemez, M., Tekinerdogan, B., & Tarhan, A. K. (2024). Microservice reference architecture design: A multi-case study. *Software: Practice and Experience*, 54(1), 58-84. <https://doi.org/10.1002/spe.3241>
- Capolupo, A., Monterisi, C., Saponieri, A., Addona, F., Damiani, L., Archetti, R., & Tarantino, E. (2021). An Interactive WebGIS Framework for Coastal Erosion Risk Management. *Journal of Marine Science and Engineering*, 9(6), 567. <https://doi.org/10.3390/jmse9060567>
- 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>
- Kothapalli, S. R. I. N. I. K. H. I. T. A., Nizamuddin, M., Talla, R. R., & Gummadi, J. C. S. (2024). DevOps and Software Architecture: Bridging the Gap between Development and Operations. *American Digits: Journal of Computing and Digital Technologies*, 2(1), 51-64.
- Lai, H., Wang, D., & Ou, X. (2023). Cross-cultural adaptation of Chinese students in the United States: Acculturation strategies, sociocultural, psychological, and academic adaptation. *Frontiers in psychology*, 13, 924561. <https://doi.org/10.3389/fpsyg.2022.924561>
- Coutts, E., Wodehouse, A., Berglund, A., Grierson, H., Liikkanen, J., Parkkamäki, H., ... & Petrakis, K. (2024). Experience of globally distributed design teams in a changing world. In *DS 131: Proceedings of the International Conference on Engineering and Product Design Education (E&PDE 2024)* (pp. 663-668). <https://doi.org/10.35199/EPDE.2024.112>
- Prenner, N., Unger-Windeler, C., & Schneider, K. (2021). Goals and challenges in hybrid software development approaches. *Journal of Software: Evolution and Process*, 33(11), e2382. <https://doi.org/10.1002/smr.2382>
- Rongali, L. P. (2022). Fostering Collaboration and Shared Ownership in Globally Distributed DevOps Teams: Challenges and Best Practices. *European Journal of Advances in Engineering and Technology*, 9(6), 96-102.

9. Cai, S. (2024). Agile project management in virtual software development teams: challenges, benefits, and implementation.
10. Dimov, A., Emanuilov, S., Bontchev, B., Dankov, Y., & Papapostolu, T. (2022). Architectural Approaches to Overcome Challenges in the Development. *Human Factors in Software and Systems Engineering*, 61, 38-43.
11. Kovalchuk, A. (2025). Complex model of business consulting for small and medium-sized enterprises: Theory, methodology and practice of implementation. Kyiv: Internauka Publishing House, 90 p. <https://doi.org/10.25313/Kovalchuk-Monograph-2025-90>