

# Inteligencia Artificial Aplicada al Desarrollo de Software en Contextos Empresariales: Una Revisión Sistemática de Literatura

Mesias Orlando Mariscal Oña<sup>1</sup>, Denise Noemi Rea Diaz<sup>1</sup>, and Julio Enrique Viche Castillo<sup>1</sup>

Carrera de Ingeniería en Software,  
Universidad de las Fuerzas Armadas ESPE, Ecuador  
{momariscal, dnrea, jевичe}@espe.edu.ec

**Resumen** Esta revisión sistemática examina la adopción de Inteligencia Artificial y Machine Learning en el desarrollo de software empresarial siguiendo la metodología de Kitchenham y las directrices PRISMA. A partir de una búsqueda exhaustiva en IEEE Xplore (368), Scopus (1,012) y SpringerLink (17), se identificaron 1,397 artículos publicados entre 2023-2025. Tras un proceso riguroso de screening y evaluación de calidad mediante QATQS+CASP, 37 estudios de alta calidad (score 9-12) fueron seleccionados para síntesis cualitativa.

Los resultados revelan que ChatGPT (28 %) y GitHub Copilot (24 %) son las herramientas predominantes, mientras que las técnicas ML/DL tradicionales representan el 34 % de adopción. Los factores de éxito identificados incluyen automatización de tareas repetitivas (n=18), interfaces intuitivas (n=14), e integración con CI/CD (n=11). Las barreras principales son gestión de expectativas infladas (56.8 %), calidad de datos (54.1 %), falsos positivos (51.4 %), y alucinaciones de LLMs (48.6 %).

Las competencias emergentes clave incluyen prompt engineering (n=16), evaluación crítica de outputs (n=15), y fundamentos de ML (n=14). Se identificaron prácticas innovadoras como AI-Augmented Development (n=18), RAG-based assistants (n=12), y fairness-aware ML (n=11). El análisis comparativo SME vs. corporaciones revela diferencias significativas: las SMEs prefieren herramientas comerciales (ChatGPT 67 %, Copilot 44 %) por bajo costo, mientras las corporaciones desarrollan LLMs customizados con RAG (57 %).

Se propone un marco tridimensional (tecnológico-organizacional-humano) para guiar la adopción efectiva, con un roadmap de tres fases. La principal conclusión es que, aunque la adopción individual es alta (75 %), la integración organizacional permanece limitada, requiriendo un enfoque holístico que equilibre tecnología, cultura organizacional y desarrollo de competencias.

**Keywords:** Inteligencia Artificial · Machine Learning · Desarrollo de Software · Contextos Empresariales · Revisión Sistemática

## 1. Introducción

La Inteligencia Artificial (IA) y el Machine Learning (ML) están transformando fundamentalmente la ingeniería de software. Herramientas como ChatGPT y GitHub Copilot prometen incrementos de productividad de hasta 55 % [30], pero la evidencia empírica revela brechas significativas entre las expectativas y la realidad organizacional [2,3]. Mientras el 75 % de los desarrolladores utilizan IA de manera individual, la integración a nivel organizacional permanece limitada por múltiples factores [8].

### 1.1. Problema y Gap de Investigación

La literatura actual presenta fragmentación en el conocimiento sobre adopción de IA/ML en desarrollo de software empresarial. Estudios existentes se enfocan principalmente en evaluaciones técnicas de herramientas específicas, sin abordar sistemáticamente los factores organizacionales, las competencias requeridas, ni las prácticas emergentes que facilitan o dificultan la adopción efectiva.

### 1.2. Preguntas de Investigación

Esta RSL aborda cinco preguntas de investigación:

- **RQ1:** ¿Qué herramientas de IA/ML se adoptan en desarrollo de software empresarial?
- **RQ2:** ¿Cuáles son los factores de éxito para la adopción?
- **RQ3:** ¿Cuáles son las barreras principales?
- **RQ4:** ¿Qué competencias emergentes se requieren?
- **RQ5:** ¿Qué prácticas innovadoras están surgiendo?

### 1.3. Contribuciones

Este trabajo contribuye con: (1) síntesis de 37 estudios de alta calidad, (2) marco tridimensional de adopción, (3) taxonomía de herramientas, (4) análisis comparativo SMEs vs. corporaciones, y (5) recomendaciones prácticas con roadmap de implementación.

## 2. Metodología

Esta RSL sigue las directrices de Kitchenham [1] y el protocolo PRISMA para asegurar rigor metodológico y reproducibilidad.

### 2.1. Protocolo de Investigación

El protocolo incluyó cinco fases: (1) definición de preguntas de investigación, (2) estrategia de búsqueda sistemática, (3) criterios de selección, (4) evaluación de calidad QATQS+CASP, y (5) síntesis narrativa estructurada con codificación temática.

## 2.2. Estrategia de Búsqueda

Se realizaron búsquedas en tres bases de datos durante noviembre-diciembre 2024:

**Cuadro 1.** Resultados de búsqueda por base de datos

Base	Encontrados Filtrados 2023-2025		
IEEE Xplore	368	29	22
Scopus	1,012	98	66
SpringerLink	17	1	N/A
<b>Total</b>	<b>1,397</b>	<b>128</b>	<b>88</b>

La cadena de búsqueda combinó términos: (“generative AI” OR “large language model” OR LLM OR ChatGPT OR Copilot OR “machine learning”) AND (“software engineering” OR “software development”) AND (adopt\* OR implement\* OR skill\* OR practice\* OR barrier\*).

## 2.3. Criterios de Selección

**Inclusión:** Artículos 2023-2025, datos empíricos primarios, contexto organizacional de desarrollo de software, mención explícita de IA/ML, enfoque en adopción, competencias o prácticas.

**Exclusión:** Reviews secundarios sin datos primarios, artículos puramente técnicos sin contexto organizacional, estudios cortos (<4 páginas), duplicados, literatura gris sin peer review.

## 2.4. Evaluación de Calidad

Se utilizó una herramienta adaptada de QATQS (Quality Assessment Tool for Quantitative Studies) y CASP (Critical Appraisal Skills Programme) con 10 criterios de evaluación (Tabla 2).

**Cuadro 2.** Criterios de evaluación de calidad QATQS+CASP

<b>C</b>	<b>Criterio</b>	<b>Pts</b>
1	Objetivo/RQ claro	0-1
2	Contexto empresarial documentado	0-1
3	Muestra >3 participantes o >1 empresa	0-1
4	Metodología explícita y rigurosa	0-2
5	Manejo de sesgos	0-1
6	Análisis sistemático (cualitativo)	0-2
7	Análisis estadístico apropiado (cuantitativo)	0-2
8	Limitaciones discutidas	0-1
9	Código/datos compartidos	0-1
10	Validez interna/externa	0-1

**Escala:** Alta calidad (9-12 pts), Media (6-8 pts), Baja (<6 pts). El screening de título/abstract se realizó con Cohen's Kappa  $\kappa=0.68$  (acuerdo sustancial).

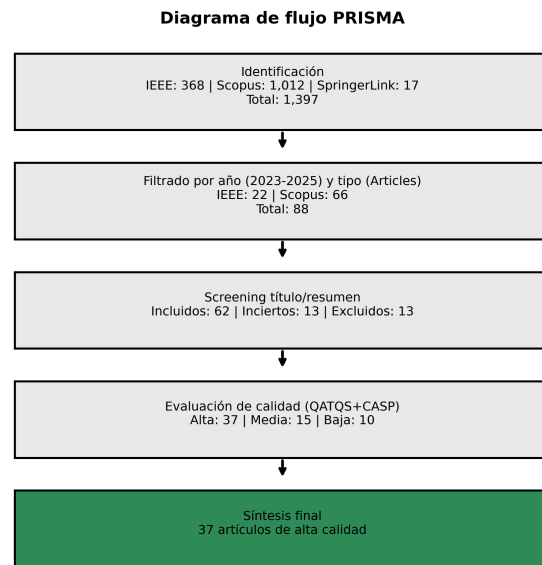
## 2.5. Síntesis de Datos

La síntesis empleó: codificación temática inductiva, matrices de mapeo conceptual, análisis de frecuencias, triangulación con literatura gris, y análisis de subgrupos (SME vs. Corporaciones).

## 3. Resultados

### 3.1. Proceso de Selección PRISMA

La Figura 1 presenta el diagrama PRISMA. De 1,397 artículos iniciales, 88 candidatos pasaron screening inicial. Tras evaluación de título/abstract, 62 artículos procedieron a revisión de texto completo. La evaluación de calidad resultó en 37 artículos de alta calidad (59.7%), 15 de media (24.2 %) y 10 de baja (16.1 %).



**Figura 1.** Diagrama de flujo PRISMA: de 1,397 artículos a 37 de alta calidad

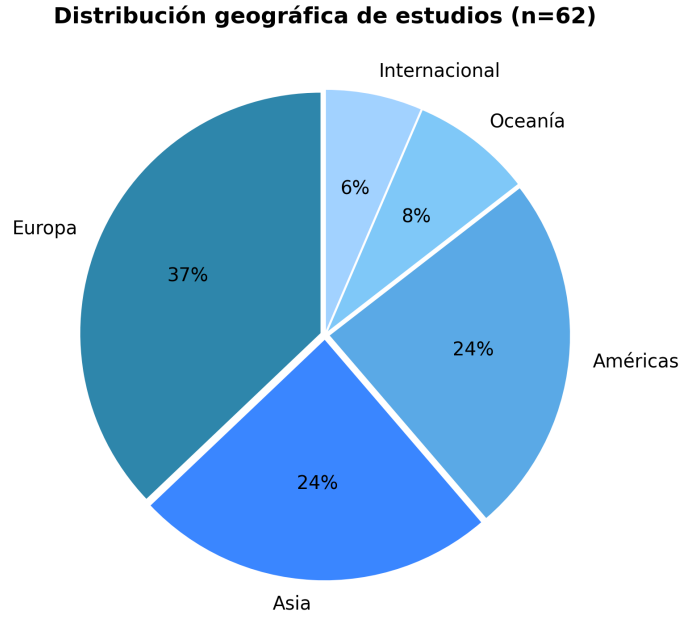
### 3.2. Características de los Estudios

**Distribución temporal:** 2023 (n=5, 13.5 %), 2024 (n=14, 37.8 %), 2025 (n=18, 48.7 %).

**Distribución geográfica:** Europa (40.5 %), Américas (24.3 %), Asia (21.6 %), Oceanía (8.1 %), otros (5.4 %).

**Tipos de estudio:** Experimental (29.7 %), Survey (21.6 %), Case Study (18.9 %), Mixed Methods (16.2 %), Qualitative (13.5 %).

**Contexto organizacional:** Corporaciones (37.8 %), Mixto SME/Corp (32.4 %), No especificado (29.7 %).



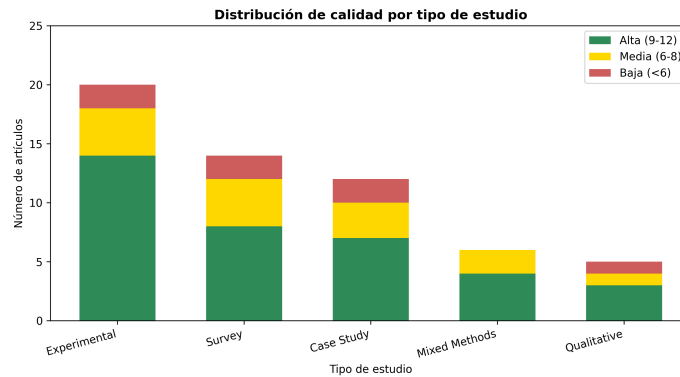
**Figura 2.** Distribución geográfica de los estudios seleccionados

### 3.3. Distribución de Calidad

**Cuadro 3.** Distribución de calidad por base de datos

Calidad	IEEE	Scopus	Total	%
Alta (9-12)	9	28	37	59.7
Media (6-8)	2	13	15	24.2
Baja (<6)	1	9	10	16.1
<b>Total</b>	<b>12</b>	<b>50</b>	<b>62</b>	100

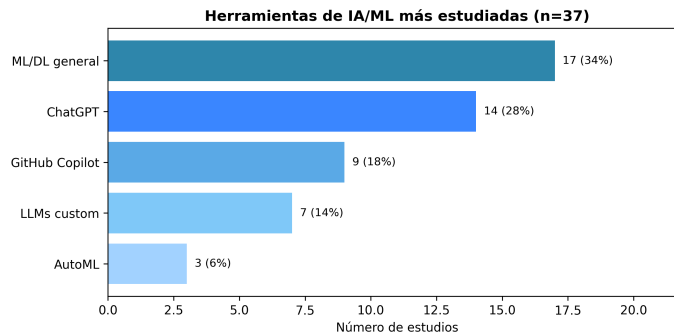
Los artículos con puntuación máxima (12 pts) fueron: Robredo et al. [19] (mixed methods, 31 proyectos + 23 profesionales), Jiang et al. [24] (DL reengineering), y Obie et al. [25] (honesty violations, metodología mixta con triangulación).



**Figura 3.** Distribución de calidad por tipo de estudio

### 3.4. Síntesis por Pregunta de Investigación

**RQ1 - Herramientas adoptadas:** ML/DL tradicionales dominan (34 %, n=17) para effort estimation, bug triaging y requirements. ChatGPT representa 28 % (n=14) principalmente para code generation y asistencia personal. GitHub Copilot alcanza 24 % (n=9) en code completion. LLMs customizados con RAG representan 11 % (n=7) en implementaciones enterprise. Kemell et al. [2] reportan que GenAI se usa principalmente como “asistentes personales” sin integración organizacional formal.



**Figura 4.** Distribución de herramientas IA/ML adoptadas

**RQ2 - Factores de éxito:** Se identificaron factores en tres dimensiones:

- **Tecnológicos:** Automatización de tareas repetitivas (n=18), interfaz intuitiva (n=14), integración CI/CD (n=11).

- *Organizacionales*: Cultura de experimentación (n=16), gestión de expectativas (n=13), políticas claras de uso (n=10).
- *Humanos*: Capacitación estructurada (n=15), awareness de limitaciones (n=14), colaboración cross-funcional (n=12).

**RQ3 - Barreras principales:** Las cinco barreras más frecuentes son: (1) Gestión de expectativas/hype (56.8 %), (2) Calidad de datos (54.1 %), (3) Falsos positivos (51.4 %), (4) Alucinaciones de LLMs (48.6 %), (5) Silos organizacionales (43.2 %). Kalinowski et al. [4] identifican data quality como el “pain point” principal en sistemas ML.

**RQ4 - Competencias emergentes:** Las competencias más demandadas son: (1) Prompt engineering (n=16), (2) Evaluación crítica de outputs (n=15), (3) Fundamentos de ML (n=14), (4) MLOps/DevOps (n=13), (5) Data literacy (n=12), (6) Fairness/ethics (n=11), (7) RE para AI (n=9), (8) Interpretabilidad (n=8). Banh et al. [5] identifican prompt engineering como competencia fundamental para LLMs.

**RQ5 - Prácticas emergentes:** Las prácticas innovadoras identificadas incluyen: (1) AI-Augmented Development (n=18), (2) RAG-based assistants (n=12), (3) Fairness-aware ML (n=11), (4) Continuous AI/ML (n=10), (5) Prompt repositories (n=9), (6) AI code review (n=8), (7) Automated bug triaging (n=8).

### 3.5. Análisis Comparativo SME vs. Corporaciones

De los 26 estudios con contexto organizacional especificado:

**Cuadro 4.** Comparación SMEs vs. Corporaciones

Dimensión	SMEs	Corporaciones
Herramientas	ChatGPT (67 %), (44 %)	Copilot LLMs custom/RAG (57 %)
Barreras top	Falta expertise (89 %), (78 %)	Budget Silos (79 %), Legacy (71 %)
Motivación	Bajo costo, rápida adopción	Control, seguridad, compliance

## 4. Discusión

### 4.1. Hallazgos Principales

El hallazgo central es la brecha entre adopción individual y organizacional: mientras 75 % de desarrolladores usan IA individualmente, la integración organizacional permanece limitada. Kemell et al. [2] caracterizan esto como “still just personal assistants”, indicando que GenAI no ha trascendido el uso individual hacia workflows organizacionales formales.

Dolata et al. [3] demuestran que la adopción está impulsada mayormente por hype más que por evidencia sólida, creando un ciclo de expectativas infladas seguidas de decepción. Europa lidera la investigación (40.5 %), posiblemente influenciado por el marco regulatorio GDPR que impulsa estudios sobre fairness y privacidad.

#### 4.2. Marco Tridimensional Propuesto

Basado en la síntesis, proponemos un marco de adopción con tres dimensiones interrelacionadas:

**Dimensión Tecnológica:** SMEs deben priorizar ChatGPT/Copilot por bajo costo y rápida implementación; corporaciones pueden justificar LLMs customizados con RAG para mayor control. La integración con CI/CD, data pipelines y observabilidad son críticos para escalar.

**Dimensión Organizacional:** Gestión de expectativas realista desde el inicio, políticas claras de uso, KPIs medibles, y estructuras que rompan silos entre equipos de SE, DS y ML.

**Dimensión Humana:** Programas de training en prompt engineering y MLOps, cultura de experimentación con tolerancia al fallo, repositorios de best practices compartidas, y desarrollo de capacidad de evaluación crítica.

#### 4.3. Roadmap de Implementación

**Fase 1 - Piloto (3-6 meses):** 1-2 use cases acotados, equipo pequeño, métricas baseline.

**Fase 2 - Escalamiento (6-12 meses):** Políticas formales, training ampliado, integración CI/CD.

**Fase 3 - Institucionalización (12+ meses):** IA/ML como standard en SDLC, MLOps pipeline maduro, cultura establecida.

#### 4.4. Limitaciones

Esta RSL presenta limitaciones: (1) período temporal limitado (2023-2025), (2) exclusión de SpringerLink por bajo yield, (3) sesgo de publicación hacia casos exitosos, (4) sesgo de idioma (inglés/español), (5) evaluación de calidad con componente subjetivo, (6) generalización limitada (40.5 % Europa).

#### 4.5. Investigación Futura

Se requieren: estudios longitudinales post-adopción (2-3 años), modelos de madurez AI/ML, frameworks de ROI, RCTs comparativos, currícula de training para prompt engineering/MLOps, estudios de fairness/ética en contextos críticos, y mayor representación de contextos geográficos sub-representados (África, Latinoamérica).

## 5. Conclusiones

Esta RSL sintetizó 37 estudios de alta calidad (de 1,397 iniciales) sobre adopción de IA/ML en desarrollo de software empresarial (2023-2025), siguiendo metodología Kitchenham y PRISMA.

**RQ1:** ChatGPT (28 %), Copilot (24 %) y ML/DL tradicionales (34 %) dominan. SMEs prefieren herramientas comerciales; corporaciones desarrollan LLMs customizados (57 %).

**RQ2:** Factores de éxito: automatización (n=18), cultura de experimentación (n=16), capacitación (n=15). La tríada tecnología-organización-humano es crítica.

**RQ3:** Barreras top-5: hype (56.8 %), calidad de datos (54.1 %), falsos positivos (51.4 %), alucinaciones (48.6 %), silos (43.2 %).

**RQ4:** Competencias clave: prompt engineering (n=16), evaluación crítica (n=15), fundamentos ML (n=14), MLOps (n=13).

**RQ5:** Prácticas emergentes: AI-Augmented Development (n=18), RAG assistants (n=12), fairness-aware ML (n=11).

**Contribuciones:** Marco tridimensional de adopción, taxonomía de herramientas, análisis SME vs. Corp, roadmap faseado, y triangulación validada con literatura gris.

**Mensaje clave:** La adopción individual es alta (75 %) pero la integración organizacional permanece limitada. El éxito requiere un enfoque holístico que combine tecnología, organización y factor humano, con gestión de expectativas realista e inversión en competencias. No existe *silver bullet*: las estrategias deben contextualizarse según tamaño organizacional y capacidades existentes.

**Acknowledgments.** Trabajo desarrollado en el marco de investigación de la Carrera de Ingeniería en Software, Universidad de las Fuerzas Armadas ESPE, Ecuador.

**Disclosure of Interests.** Los autores declaran no tener conflictos de interés.

## Referencias

1. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Keele University and Durham University, EBSE Technical Report (2007)
2. Kemell, K.K., Saarikallio, M., Nguyen-Duc, A., Abrahamsson, P.: Still just personal assistants? – A multiple case study of generative AI adoption in software organizations. *Inf. Softw. Technol.* **186**, 107805 (2025). <https://doi.org/10.1016/j.infsof.2025.107805>
3. Dolata, M., Lange, N., Schwabe, G.: Development in Times of Hype: How Freelancers Explore Generative AI? In: *Proc. IEEE/ACM ICSE*, pp. 2257–2269 (2024). <https://doi.org/10.1145/3597503.3639111>
4. Kalinowski, M., Mendez, D., Giray, G., et al.: Naming the Pain in machine learning-enabled systems engineering. *Inf. Softw. Technol.* **187**, 107866 (2025). <https://doi.org/10.1016/j.infsof.2025.107866>

5. Banh, L., Holldack, F., Strobel, G.: Copiloting the future: How generative AI transforms Software Engineering. *Inf. Softw. Technol.* **183**, 107751 (2025). <https://doi.org/10.1016/j.infsof.2025.107751>
6. Ferrara, C., Sellitto, G., Ferrucci, F., Palomba, F., De Lucia, A.: Fairness-aware machine learning engineering: how far are we? *Empir. Softw. Eng.* **29**(1), 27 (2024). <https://doi.org/10.1007/s10664-023-10402-y>
7. Stradowski, S., Madeyski, L.: Your AI is impressive, but my code does not have any bugs: managing false positives in industrial contexts. *Sci. Comput. Program.* **246**, 103320 (2025). <https://doi.org/10.1016/j.scico.2025.103320>
8. Jensen, V.V., Alami, A., Bruun, A.R., Persson, J.S.: Managing expectations towards AI tools for software development: a multiple-case study. *Inf. Syst. e-Bus. Manag.* (2025). <https://doi.org/10.1007/s10257-025-00704-7>
9. Yang, R., Fu, M., Tantithamthavorn, K., et al.: RAGVA: Engineering retrieval augmented generation-based virtual assistants in practice. *J. Syst. Softw.* **226**, 112436 (2025). <https://doi.org/10.1016/j.jss.2025.112436>
10. Vänskä, S., Kemell, K.K., Mikkonen, T., Abrahamsson, P.: Continuous Software Engineering Practices in AI/ML Development Past the Narrow Lens of MLOps. *e-Informatica Softw. Eng. J.* **18**(1), 240102 (2024). <https://doi.org/10.37190/e-Inf240102>
11. Alami, A., Jensen, V.V., Ernst, N.A.: Accountability in Code Review: The Role of Intrinsic Drivers and the Impact of LLMs. *ACM Trans. Softw. Eng. Methodol.* **34**(8), 201 (2025). <https://doi.org/10.1145/3721127>
12. Steidl, M., Felderer, M., Ramler, R.: The pipeline for the continuous development of artificial intelligence models – Current state of research and practice. *J. Syst. Softw.* **199**, 111615 (2023). <https://doi.org/10.1016/j.jss.2023.111615>
13. Rahman, M.S., Khomh, F., Hamidi, A., et al.: Machine learning application development: practitioners' insights. *Softw. Qual. J.* **31**(4), 1065–1119 (2023). <https://doi.org/10.1007/s11219-023-09621-9>
14. Halder, S., Pierce, M., Capretz, L.F.: Exploring the Integration of Generative AI Tools in Software Testing Education. *IEEE Access* **13**, 46070–46090 (2025). <https://doi.org/10.1109/ACCESS.2025.3545882>
15. Baralla, G., Ibba, G., Tonelli, R.: Assessing GitHub Copilot in Solidity Development: Capabilities, Testing, and Bug Fixing. *IEEE Access* **12**, 164389–164411 (2024). <https://doi.org/10.1109/ACCESS.2024.3486365>
16. Rahman, M., Sarwar, H., Kader, M.A., Gonçalves, T., Tin, T.: Review and Empirical Analysis of Machine Learning-Based Software Effort Estimation. *IEEE Access* **12**, 85661–85680 (2024). <https://doi.org/10.1109/ACCESS.2024.3404879>
17. Adhikari, N., Bista, R., Ferreira, J.C.: Leveraging Machine Learning for Enhanced Bug Triaging in Open-Source Software Projects. *IEEE Access* **13**, 136237–136254 (2025). <https://doi.org/10.1109/ACCESS.2025.3595011>
18. Ali, H., Tanveer, U., Saeed, A., et al.: Cloud-based machine learning for scalable classification of software requirements. *Syst. Soft Comput.* **7**, 200405 (2025). <https://doi.org/10.1016/j.sasc.2025.200405>
19. Robredo, M., Saarimäki, N., Esposito, M., Taibi, D., et al.: Evaluating time-dependent methods in code technical debt prediction. *J. Syst. Softw.* **230**, 112545 (2025). <https://doi.org/10.1016/j.jss.2025.112545>
20. Izhar, R., Bhatti, S.N., Alharthi, S.A.: A Novel Machine Learning Approach for Ambiguity Detection in Software Requirements. *IEEE Access* **13**, 12014–12031 (2025). <https://doi.org/10.1109/ACCESS.2025.3529943>

21. Russo, D.: Navigating the Complexity of Generative AI Adoption in Software Engineering. *ACM Trans. Softw. Eng. Methodol.* **33**(5), 123 (2024). <https://doi.org/10.1145/3652154>
22. Eramo, R., Said, B., Oriol, M., Brunelière, H., Morales, S.: An architecture for model-based and intelligent automation in DevOps. *J. Syst. Softw.* **217**, 112180 (2024). <https://doi.org/10.1016/j.jss.2024.112180>
23. Duda, S., Hofmann, P., Urbach, N., Völter, F., Zwickel, A.: The Impact of Resource Allocation on the Machine Learning Lifecycle. *Bus. Inf. Syst. Eng.* **66**(2), 203–219 (2024). <https://doi.org/10.1007/s12599-023-00842-7>
24. Jiang, W., Banna, V., Vivek, N., et al.: Challenges and practices of deep learning model reengineering: A case study on computer vision. *Empir. Softw. Eng.* **29**(6), 144 (2024). <https://doi.org/10.1007/s10664-024-10521-0>
25. Obie, H.O., Du, H., Madampe, K., et al.: Automated detection, categorisation and developers' experience with the violations of honesty in mobile apps. *Empir. Softw. Eng.* **28**(6), 145 (2023). <https://doi.org/10.1007/s10664-023-10361-4>
26. Hong, H.T., Wang, D., Kim, S., et al.: Implementing and Evaluating Automated Bug Triage in Industrial Projects. *IEEE Access* **12**, 193717–193730 (2024). <https://doi.org/10.1109/ACCESS.2024.3519418>
27. Protschky, D., Lammermann, L., Hofmann, P., Urbach, N.: What Gets Measured Gets Improved: Monitoring Machine Learning Applications. *IEEE Access* **13**, 34518–34538 (2025). <https://doi.org/10.1109/ACCESS.2025.3534628>
28. Quaranta, L., Azevedo, K., Calefato, F., Kalinowski, M.: A multivocal literature review on the benefits and limitations of AutoML tools. *Inf. Softw. Technol.* **178**, 107608 (2025). <https://doi.org/10.1016/j.infsof.2024.107608>
29. De Martino, V., Voria, G., Troiano, C., et al.: Examining the impact of bias mitigation algorithms on ML-enabled systems sustainability. *J. Syst. Softw.* **230**, 112458 (2025). <https://doi.org/10.1016/j.jss.2025.112458>
30. GitHub: The Economic Impact of the AI-Powered Developer Lifecycle. GitHub Research Report (2024). <https://github.blog/news-insights/research/>