

Técnicas y Aplicaciones de Inteligencia Artificial en el Ciclo de Vida de Desarrollo de Software

Jose-David Pérez¹, Maria Monserrat¹, Antonio Crespi¹, Antoni-Lluís Mesquida¹, and Antònia Mas¹

Universidad de las Islas Baleares, Palma de Mallorca, España
josedavidperezcanellas@gmail.com

{maria.monserrat, antonio.crespi, antoni.mesquida, antonia.mas}@uib.cat

Resumen El uso de aplicaciones y técnicas de Inteligencia Artificial (IA) en las distintas fases del ciclo de vida del desarrollo de software (SDLC) busca mejorar la eficiencia, la precisión y la innovación en cada etapa del proceso. Este artículo presenta una investigación en curso que nace con el propósito de resumir las técnicas y aplicaciones de IA aplicadas en las diferentes fases del SDLC: planificación, análisis, diseño, implementación, pruebas, despliegue y mantenimiento. El objetivo de esta investigación es identificar las tareas del SDLC que pueden ser automatizadas total o parcialmente mediante IA, detectar posibles lagunas en la automatización dentro de la ingeniería del software y proporcionar una base de conocimiento para investigadores y profesionales interesados en aprovechar la IA para transformar los flujos de trabajo en esta disciplina.

Keywords: Técnicas y Aplicaciones · Inteligencia Artificial · Ciclo de Vida del Desarrollo de Software

1. Introducción

Los recientes avances en IA han permitido la automatización de tareas repetitivas y mejorado la toma de decisiones en diversos ámbitos. Su aplicación abarca desde la redacción de documentos, el procesamiento de datos, la traducción automática, el análisis de imágenes y la optimización de procesos industriales. Estas capacidades han demostrado mejorar la eficiencia en una variedad de ámbitos, impulsando el desarrollo de nuevas herramientas y metodologías.

En el caso particular de la ingeniería del software, la IA ha sido utilizada para mejorar distintas fases del SDLC. En particular, se ha observado un crecimiento en el uso de modelos de aprendizaje profundo, como GitHub Copilot, y de entornos de desarrollo basados en IA, como IDX, Codium y Cursor, en tareas de generación de código, detección de vulnerabilidades y asistencia técnica durante el desarrollo de software [1]. Se prevé que el número de estudios en este área continúe en aumento, gracias a los beneficios de la IA frente a los enfoques tradicionales.

Este estudio se centra en el análisis de la aplicación y técnicas de IA en procesos de desarrollo de software. En este contexto, presentamos una propuesta de investigación cuyo objetivo es definir un conjunto de técnicas y aplicaciones de IA que faciliten la automatización de tareas dentro de cada fase del SDLC. Consideramos una técnica como un método o algoritmo de IA, y una aplicación como su uso práctico mediante una interfaz de usuario para tareas de ingeniería del software. Se espera que esta investigación proporcione un marco de referencia detallado sobre el uso de técnicas y aplicaciones de IA en cada proceso del SDLC. Asimismo, busca identificar qué tareas pueden beneficiarse de la automatización mediante IA, analizar tendencias en la adopción de estas tecnologías y detectar posibles áreas de mejora o investigación futura en la integración de IA en la ingeniería del software.

2. Resultados preliminares

Para llevar a cabo esta investigación, se ha iniciado una revisión sistemática de la literatura con el fin de identificar y analizar el uso de técnicas y aplicaciones de IA en las diferentes fases del SDLC. La metodología sigue el enfoque propuesto por Kitchenham y Charters [2] para revisiones sistemáticas en ingeniería del software. En primer lugar, se han formulado las siguientes preguntas de investigación:

- ¿Cuáles son las principales técnicas de IA aplicadas en cada fase del SDLC?
- ¿Cuáles son las aplicaciones más utilizadas en la automatización de tareas dentro del SDLC?
- ¿Cuál ha sido la evolución temporal de la aplicación de IA en la ingeniería del software?

A continuación, se ha establecido la consulta de búsqueda utilizando los siguientes términos clave para garantizar una cobertura amplia y precisa de la literatura relevante: (*software life cycle OR software lifecycle OR software development OR software stages OR software phases*) AND (*artificial intelligence tool OR artificial intelligence application OR artificial intelligence app OR artificial intelligence software*)

La búsqueda se ha realizado en bases de datos científicas como IEEE Xplore, ACM Digital Library y Scencedirect, Springerlink, Wiley Online Library y Taylor & Francis. Posteriormente, se han aplicado criterios de inclusión y exclusión para la selección de estudios relevantes. Tras el cribado, se lleva a cabo un proceso de extracción y síntesis de datos con el propósito de identificar tendencias, lagunas y oportunidades en la integración de IA en el SDLC.

Los resultados preliminares apuntan a que la mayoría de aplicaciones de IA son para la fase de implementación y de pruebas, principalmente para las tareas de generación de código [3] y generación de tests [4]. Por el contrario, en las fases de planificación y análisis se han encontrado menos estudios, lo que sugiere unas áreas de investigación poco exploradas.

Respecto a las técnicas de IA, los resultados preliminares apuntan a que la mayoría son para la fase de pruebas, principalmente para las tareas de predicción de defectos y de detección de vulnerabilidades [5,6]. Estas técnicas han demostrado mejorar la calidad del software previniendo defectos y vulnerabilidades en la fase de pruebas.

En los últimos años parece haber una tendencia creciente en el número de trabajos que abordan el estudio de aplicaciones de IA, respecto a los estudios de técnicas de IA. De los cuales, la mayoría de estas aplicaciones de IA son chatbots o asistentes conversacionales [7] basados en IA Generativa, lo que refleja un interés creciente para la asistencia y automatización de las tareas de desarrollo y soporte técnico en el desarrollo de software.

3. Primeras apreciaciones y trabajo futuro

Con el objetivo de analizar el impacto de las aplicaciones y técnicas de IA en los diferentes procesos del SDLC, este artículo presenta una revisión sistemática de la literatura en curso. Se pretende identificar qué tareas pueden ser automatizadas total o parcialmente mediante IA, detectar áreas aún no exploradas en la automatización de procesos dentro de la ingeniería del software y ofrecer un marco de referencia que facilite su adopción.

Los resultados preliminares indican que la mayoría de las aplicaciones de IA están dirigidas a la fase de implementación, con herramientas enfocadas en la generación y sugerencia de código. Asimismo, en la fase de pruebas, se observa un incremento en la adopción de IA para la generación de tests y la detección de vulnerabilidades. Sin embargo, se han encontrado menos estudios sobre la aplicación de IA en las fases iniciales del SDLC, como la planificación y el análisis, y en la fase de despliegue.

Se espera que la investigación proporcione un análisis detallado del uso de IA en cada fase del SDLC, identificando tendencias y posibles oportunidades para su integración. Para abordar este trabajo de investigación, se plantea una estrategia en cuatro fases. Esta estrategia incluye la recopilación y el análisis de estudios previos, la clasificación de herramientas según su funcionalidad y grado de automatización, y la evaluación de su impacto en la ingeniería del software; el desarrollo de un repositorio abierto para facilitar el acceso a todas las aplicaciones de IA, agrupadas por diferentes categorías; la evaluación empírica de aplicaciones de IA generativa, y su correspondiente medición de eficiencia y margen de mejora, en el SDLC; y, finalmente, el desarrollo de una aplicación de IA para alguna de las fases del desarrollo de software.

Se espera que los resultados de la revisión sistemática que se realiza durante la primera fase, proporcionen una visión detallada sobre cómo las técnicas y aplicaciones de IA están siendo implementadas en cada fase del SDLC. En particular, se pretende identificar qué tareas en las fases de planificación, análisis, diseño, implementación, pruebas, despliegue y mantenimiento han sido automatizadas mediante IA, así como aquellas que aún presentan oportunidades para futuras investigaciones. Asimismo, se desea analizar la diferencia en la adopción

de técnicas de IA (como redes neuronales, algoritmos genéticos y aprendizaje por refuerzo) en comparación con aplicaciones de IA específicas (como asistentes de codificación o herramientas de generación automática de pruebas). Además, se prevé realizar un análisis de la evolución temporal de la investigación en este campo, mostrando la cantidad de trabajos publicados en técnicas y aplicaciones de IA en ingeniería del software a lo largo de los años, con el fin de identificar tendencias de estas tecnologías.

En una segunda fase, se pretende desarrollar un repositorio de herramientas para los diferentes roles implicados en el SDLC, como jefes de proyecto, analistas, desarrolladores, testers, entre otros, puedan disponer de todas estas aplicaciones en un lugar centralizado. En una tercera fase, se desea realizar una evaluación empírica de aplicaciones de IA Generativa para las diferentes fases del SDLC, con el objetivo de analizar las ventajas, limitaciones y posibles mejoras a futuro de este tipo de herramientas. Finalmente, en una cuarta fase, se plantea desarrollar una herramienta de IA que dé soporte en algunas de las fases del SDLC de las cuales se hayan detectado un mayor margen de mejora en la tercera fase.

Referencias

1. Martinović, B., Rozić, R. Perceived Impact of AI-Based Tooling on Software Development Code Quality. *SN COMPUT. SCI.* **6**, 63 (2025). <https://doi.org/10.1007/s42979-024-03608-4>
2. B. Kitchenham, "Guidelines for performing Systematic Literature Reviews in Software Engineering (Software Engineering Group, Department of Computer Science, Keele . . .," no. October 2021, 2007.
3. Gustavo Pinto, Cleidson De Souza, Thayssa Rocha, Igor Steinmacher, Alberto Souza, and Edward Monteiro. 2024. Developer Experiences with a Contextualized AI Coding Assistant: Usability, Expectations, and Outcomes. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering - Software Engineering for AI (CAIN '24)*. Association for Computing Machinery, New York, NY, USA, 81–91 <https://doi.org/10.1145/3644815.3644949>
4. A. Awad, M. H. Qutqut, A. Ahmed, F. Al-Haj and F. Almasalha, "Artificial Intelligence Role in Software Automation Testing," 2024 International Conference on Decision Aid Sciences and Applications (DASA), Manama, Bahrain, 2024, pp. 1-6, doi: <https://doi.org/10.1109/DASA63652.2024.10836630>.
5. S. Stradowski and L. Madeyski, "Can we Knapsack Software Defect Prediction? Nokia 5G Case," 2023 IEEE/ACM 45th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Melbourne, Australia, 2023, pp. 365-369, doi: <https://doi.org/10.1109/ICSE-Companion58688.2023.00104>.
6. Yusupbekov, N. R., Gulyamov, S. M., Usmanova, N. B., & Mirzaev, D. A. (2017). Challenging the ways to determine the faults in software: Technique based on associative interconnections. *Procedia Computer Science*, 120, 641–648. <https://doi.org/10.1016/J.PROCS.2017.11.290>
7. Stefano Lambiase, Gemma Catolino, Fabio Palomba, and Filomena Ferrucci. 2024. Motivations, Challenges, Best Practices, and Benefits for Bots and Conversational Agents in Software Engineering: A Multivocal Literature Review. *ACM Comput. Surv.* **57**, 4, Article 93 (April 2025), 37 pages. <https://doi.org/10.1145/3704806>