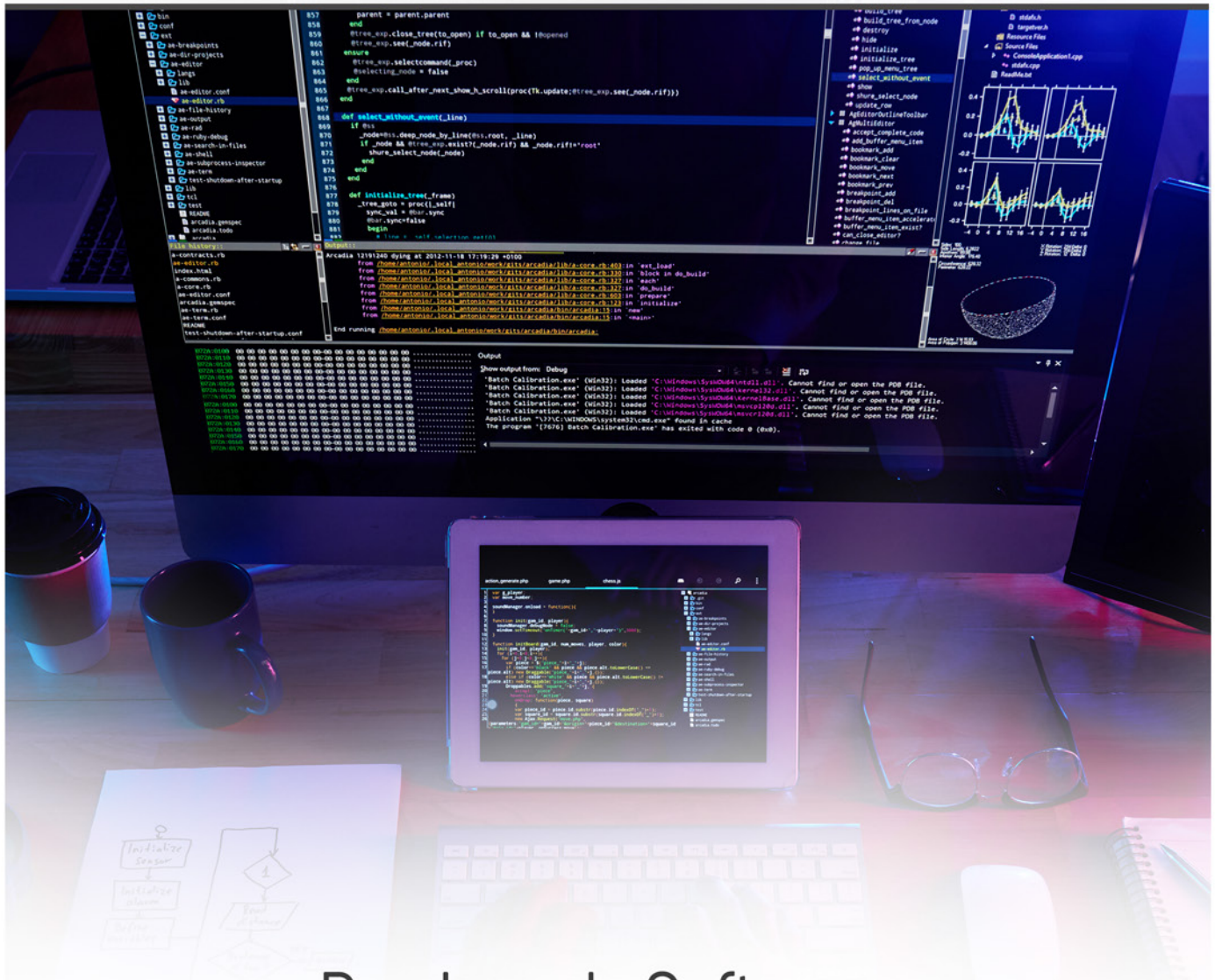




Pruebas de Software

XXXXXXXXXX
XXXXXXXXXX
XXXXXXXXXX

Autores:

Angel Felipe Pérez Serrato
Richard Blanquicett Cardozo
Yefrei Camilo Galvis Diaz
Jesus Manuel Joven Cordoba

Resumen

El desarrollo ágil y el despliegue continuo han elevado la relevancia de la calidad del software, haciendo de las pruebas un componente esencial en el ciclo de vida del desarrollo (CI/CD). Este artículo examina las tendencias emergentes en pruebas de software hacia 2025, subrayando: (a) la integración acelerada de inteligencia artificial y machine learning para generación automática de casos de prueba, autorreparación de scripts y priorización en regresión; (b) enfoques shift-left y shift-right como prácticas complementarias para asegurar calidad desde diseño hasta producción; (c) la priorización eficaz de casos de prueba mediante aprendizaje automático, incluyendo técnicas como random forest, redes neuronales y aprendizaje por refuerzo. Se discuten beneficios como mayor cobertura, eficiencia y resiliencia frente a cambios de UI, junto a riesgos derivados del exceso de velocidad sin salvaguardas, evocando la “trampa de velocidad AI” (AI speed trap). Concluimos proponiendo un enfoque equilibrado basado en automatización avanzada, supervisión humana, métricas claras y gobernanza de IA. Palabras clave: pruebas de software, inteligencia artificial, shift-left, regresión, priorización, CI/CD.

Palabras clave: Pruebas de software, Inteligencia artificial (IA), Shift-left, Shift-right, Priorización de casos de prueba, Pruebas de regresión, Automatización, Calidad del software

Abstract

The agile development and continuous deployment have elevated the relevance of software quality, making testing a core component of the CI/CD lifecycle. This paper reviews emerging trends in software testing toward 2025, emphasizing: (a) accelerated integration of artificial intelligence and machine learning for automatic test case generation, self-healing scripts, and regression prioritization; (b) complementary shift-left and shift-right approaches to ensure quality from design to production; (c) effective regression test prioritization using ML techniques, including random forests, neural networks, and reinforcement learning. We discuss benefits such as increased coverage, efficiency, and resilience to UI changes, alongside risks of speed without safeguards, referencing the AI speed trap. We conclude by proposing a balanced approach that combines advanced automation, human oversight, clear metrics, and AI governance.

Keywords: Software testing, Artificial intelligence (AI), Shift-left, Shift-right, Test case prioritization, Regression testing, Automation, Software quality

Introducción

En el amplio contexto del desarrollo de software moderno, donde es indispensable la entrega continua y la rápida repetición se ha vuelto prioritarias, la calidad de software se ha consolidado como algo clave para reducir los problemas o fallos en producción e incrementar la satisfacción del cliente. En este tipo de escenario, las pruebas de software desempeñan un rol especial y esencial, ya que permite saber y detectar los defectos, verificar requisitos y validar de forma positiva la funcionalidad del sistema en distintas etapas del ciclo de vida del software.

Uno de los temas mas destacados en la literatura reciente es la priorización y selección de unos casos de prueba en pruebas de regresión. Pan et al. (2021) realizaron una revision sistemática de 29 estudios en los que se combinan varios modelos de Machine Learning con TSP (Test-cas Selection and Prioritization) para poder optimizar el proceso de pruebas y reducir drásticamente el tiempo sin sacrificar la calidad. Tambien, Singhal et al. (2021) hicieron un “tertiary study” que logra sintetizar las principales metodologías de prueba de regresión, evidenciando la creciente adopción de enfoques automatizados en el entorno de DevOps.

También, Zardari et al. (2022) llevaron a cabo un análisis bibliométrico cubriendo casos de uso en testing entre 2016 y 2021, identificando varias tendencias resaltantes como la integración de inteligencia artificial para la priorización de pruebas y la evolución de las practicas ágiles en testing. Por otra parte, Tran et al. (2024) presentaron un estudio terciario enfocado en la evaluación de la calidad de los artefactos de prueba, proponiendo un modelo basado en atributos de calidad (tales como cobertura, coherencia y mantenimiento), alineados a los estándares como ISO/IEC 25010.

En el contexto actual del desarrollo de software —donde la entrega continua (CI/CD) es imperativa—, asegurar la **calidad** se ha convertido en un componente esencial para reducir fallos en producción y aumentar la satisfacción del usuario. Las **pruebas de software** permiten detectar defectos, verificar requisitos y validar funcionalidad en distintas fases del ciclo de vida.

Entre las tendencias más relevantes según la literatura reciente, se destaca:

- La integración de **AI y ML**, facilitando generación de casos de prueba, priorización inteligente y autorreparación de scripts automáticos [TestRail | The Quality OS for QA](#) [Teamsglobalapptesting.com](#) [aqua cloud - best software for testing](#) [Testlio](#).
- La adopción de **shift-left** (pruebas desde etapas tempranas: requisitos, diseño) y **shift-right** (pruebas en producción para feedback continuo) [TestRail | The Quality OS for QA](#) [TeamsTestingXperts](#) [Bugasuraglobalapptesting.com](#).
- La priorización en regresión mediante **aprendizaje automático**, como random forests, redes neuronales, y modelos de refuerzo, mejorando eficiencia sin sacrificar detección de errores [dergipark.org.tr/jai.front-sci.com](#) [arXiv+3arXiv+3arXiv+3](#).
- La alerta sobre la **trampa de velocidad AI** —automatización sin gobernanza puede comprometer la calidad y estabilidad del software [TechRadar](#).

Se propone en este estudio ofrecer un panorama actualizado y sistemático de estas tendencias y su impacto en la calidad de software.

Metodología

La presente investigación se desarrolló bajo un enfoque **documental y cualitativo**, orientado a la revisión y análisis de literatura reciente sobre pruebas de software en el período **2021–2025**.

Para la búsqueda y selección de fuentes se consultaron bases de datos académicas reconocidas, tales como **IEEE Xplore**, **ACM Digital Library**, **ScienceDirect** y **arXiv**, así como portales especializados en calidad de software (TechRadar, TestRail, TestingXperts, entre otros).

Los criterios de inclusión consideraron:

- Publicaciones en inglés o español entre **2021 y 2025**.
- Estudios primarios (artículos experimentales), secundarios (revisiones sistemáticas) y terciarios (meta-análisis).
- Blogs técnicos y reportes de la industria con alto nivel de citación o relevancia en tendencias emergentes.

Se excluyeron documentos sin respaldo académico o técnico verificable, así como publicaciones anteriores a 2021.

El procedimiento metodológico se llevó a cabo en tres fases:

1. **Recopilación de literatura:** identificación de 45 documentos iniciales.
2. **Filtrado y depuración:** aplicación de criterios de inclusión/exclusión, reduciendo la muestra a **25 fuentes principales**.
3. **Análisis cualitativo y comparativo:** codificación temática de los hallazgos, clasificando tendencias en tres ejes:
 - Inteligencia artificial y machine learning en pruebas de software.
 - Enfoques shift-left y shift-right.
 - Priorización de casos de prueba en regresión.

Este procedimiento permitió consolidar un panorama actualizado de las prácticas y desafíos emergentes en pruebas de software hacia 2025.



Resultados y discusión

a) AI y ML en pruebas de software

- AI genera casos de prueba (incluso generativos), autorrepara scripts (self-healing) y optimiza el mantenimiento [Bugasuraaqua cloud - best software for testingTestlio](#).
- Plataformas sin código (low/no-code) permiten que testers sin experiencia en programación puedan crear pruebas efectivas [ZohoACCELO](#).
- Riesgo: velocidad sobre calidad. El exceso de automatización sin gobernanza lleva a errores graves, vulnerabilidades y altos costos operativos [TechRadar](#).

b) Shift-left y shift-right

- Shift-left detecta errores temprano, reduciendo costos de corrección y agilizando releases [TestRail | The Quality OS for QA Teamsglobalapptesting.com](#).
- Shift-right mejora la calidad en producción, con feedback real y pruebas resilientes como ingeniería del caos [TestingXperts](#).

c) Priorización de pruebas de regresión con ML

- Modelos como random forest, redes neuronales y Markov se usan exitosamente para priorizar casos de prueba [dergipark.org.trjai.front-sci.comijisae.orgarXiv+2arXiv+2](#).
- Esta práctica reduce la cantidad de tests ejecutados mientras mantiene alta capacidad de detección de fallos.

Conclusiones

Las pruebas de software están transitando hacia un modelo más **automatizado, inteligente y continuo**, impulsado por AI/ML, y respaldado por prácticas shift-left/right. La priorización basada en ML representa una solución efectiva para ciclos de regresión más rápidos y con menor recurso. No obstante, la automatización debe estar acompañada de **gobernanza, supervisión humana** y métricas de calidad adecuadas para evitar la trampa de velocidad.

Se recomienda:

- Implementación gradual de AI/ML en el pipeline de pruebas.
- Diseño de métricas claras (resiliencia, cobertura real, estabilidad en producción).
- Equipos AI-literate con comprensión contextual sobre automatización.
- Gobernanza clara para la automatización, asegurando transparencia y trazabilidad.

Referencias Bibliográficas

- Pan, R., Bagherzadeh, M., Ghaleb, T. A., & Briand, L. (2021). *Test Case Selection and Prioritization Using Machine Learning: A Systematic Literature Review*. arXiv. [arXiv](#)
- Omri, S., & Sinz, C. (2021). *Machine Learning Techniques for Software Quality Assurance: A Survey*. arXiv. [arXiv](#)
- Yiling Lou. (s. f.). *A Survey on Regression Test-Case Prioritization*. ScienceDirect. [ScienceDirect](#)
- Anonymous. (2025). *9 Software Testing Trends for 2025*. TestRail Blog. [TestRail | The Quality OS for QA Teams](#)
- Global App Testing. (2025). *10 software testing trends that will dominate the market in 2025*. [globalapptesting.com](#)
- TestingXperts. (2025). *Top 7 Emerging Software Testing Trends That Will Dominate in 2025*. [TestingXperts](#)
- ACCELQ. (2025). *Software Testing Trends to Look Out For in 2025*. [ACCELQ](#)
- Zoho QEngine. (2025). *11 software testing trends for 2025*. [Zoho](#)
- Aqua-Cloud. (2025). *AI in Software Testing: Best Practices in 2025*. [aqua cloud - best software for testing](#)
- Testlio. (2025). *AI in Software Testing: Actionable Advice for 2025*. [Testlio](#)
- Bugasura. (2025). *The Latest Trends in Software Testing Shaping 2025 and Beyond*. [Bugasura](#)
- TechRadarPro. (2025, agosto). *The AI speed trap: why software quality is falling behind in the race to release*. [TechRadar](#)