NIGMA SOLUTIONS

Requerimientos versus calidad en DevOps

MARZO
2025

NIGMA SOLUTIONS
WWW.NIGMA.CL
GMB

Introdución

Este artículo analiza la **evolución histórica** de los requerimientos no funcionales (**RNF**) y su transformación en los fundamentos de calidad en un entorno **DevOps**. Se exploran las diferencias en enfoque, implementación, medición y cultura entre ambos paradigmas, destacando cómo la integración de prácticas ágiles y la retroalimentación continua han modificado la manera de abordar la calidad del software. Además, **se discuten las repercusiones negativas de no comprender** esta evolución, evidenciadas a través de ejemplos prácticos inspirados en **The Phoenix Project,** tales como problemas de comunicación, resistencia al cambio y la incapacidad para responder ágilmente a incidencias. El estudio concluye resaltando la importancia de una visión integral que combine la planificación estructurada de los RNF con la flexibilidad operativa de DevOps para alcanzar una mejora continua en la calidad y resiliencia del software.

Evolución y Diferencias entre los Requerimientos No Funcionales y los Fundamentos de Calidad en DevOps

En el desarrollo de software, los requerimientos no funcionales (RNF) han sido **tradicionalmente esenciales para definir características de calidad** como rendimiento, seguridad, usabilidad y escalabilidad. Con el advenimiento de metodologías ágiles y, en especial, de la cultura DevOps, ha **surgido un nuevo enfoque** para gestionar la calidad.

¿Cómo han evolucionado los RNF y en qué se diferencian de los fundamentos de calidad en DevOps?

Historia y Evolución

Requerimientos No Funcionales (RNF)

Los RNF surgieron en la ingeniería de software para **garantizar calidad en rendimiento, confiabilidad y seguridad**, además de las funciones esperadas. Estándares como IEEE 830-1998 e ISO/IEC 25010 permitieron definir y **medir estos atributos de forma estructurada.**

Definidos en las etapas iniciales del desarrollo, facilitan la validación del producto. Sin embargo, **su enfoque rígido presenta limitaciones ante las necesidades tecnológicas actuales.**

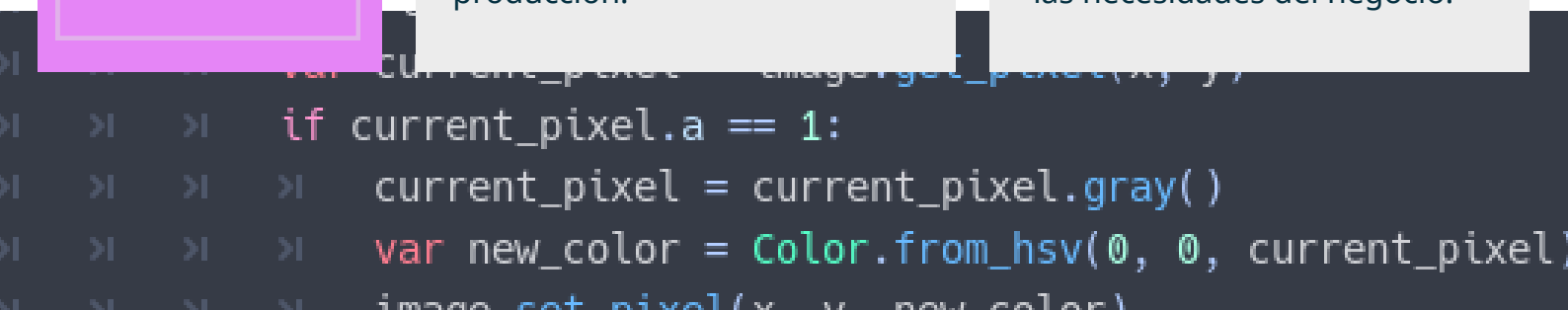
Evolución hacia DevOps

La adopción de metodologías ágiles y DevOps **transformó los modelos tradicionales**, fomentando la colaboración entre desarrollo y operaciones con integración, despliegue continuo y automatización.

La **calidad ahora se monitorea constantemente** con métricas como SLOs (Service Level Objectives), SLIs (Service Level Indicators) y SLAs (Service Level Agreements), evaluándose en producción para ajustarse rápidamente a cambios y nuevas demandas.

Diferencias entre RNF y Fundamentos de Calidad en DevOps

	RNF	DEVOPS
ENFOQUE Y PROPÓSITO	Se **documentan de forma anticipada** y establecen criterios de calidad que se deben alcanzar al final del ciclo de desarrollo. **Su carácter es relativamente estático**, ya que se fijan en fases tempranas y sirven de base para la validación final del sistema.	La **calidad se integra en todo el ciclo de vida del software**. Se centra en la observabilidad, el monitoreo continuo y la capacidad de respuesta inmediata ante incidencias. Este **enfoque es dinámico** y adaptativo, permitiendo ajustes constantes basados en feedback en tiempo real.
IMPLEMEN-TACIÓN Y MEDICIÓN	Se implementan **durante la fase de diseño** y se validan mediante pruebas de aceptación y evaluaciones estáticas en entornos controlados. Esto puede generar **discrepancias entre lo planificado y el resultado experimentado en producción aumentando los costos**	La filosofía DevOps incorpora herramientas de monitoreo y análisis en producción que permiten medir el desempeño del sistema de forma continua. Los **indicadores de calidad se actualizan en tiempo real**, facilitando la **detección temprana de desviaciones** y la aplicación de **correcciones ágiles**.
CULTURA Y RESPONSA-BILIDAD	La definición y garantía de los RNF **es responsabilidad principalmente de equipos de arquitectura y desarrollo**, lo que puede generar una **separación entre quienes diseñan el sistema y quienes operan el servicio** en producción.	Se **promueve la responsabilidad compartida** entre desarrollo y operaciones. La colaboración constante y el intercambio de información **fomentan una cultura de mejora continua y una rápida adaptación** a las necesidades del negocio.



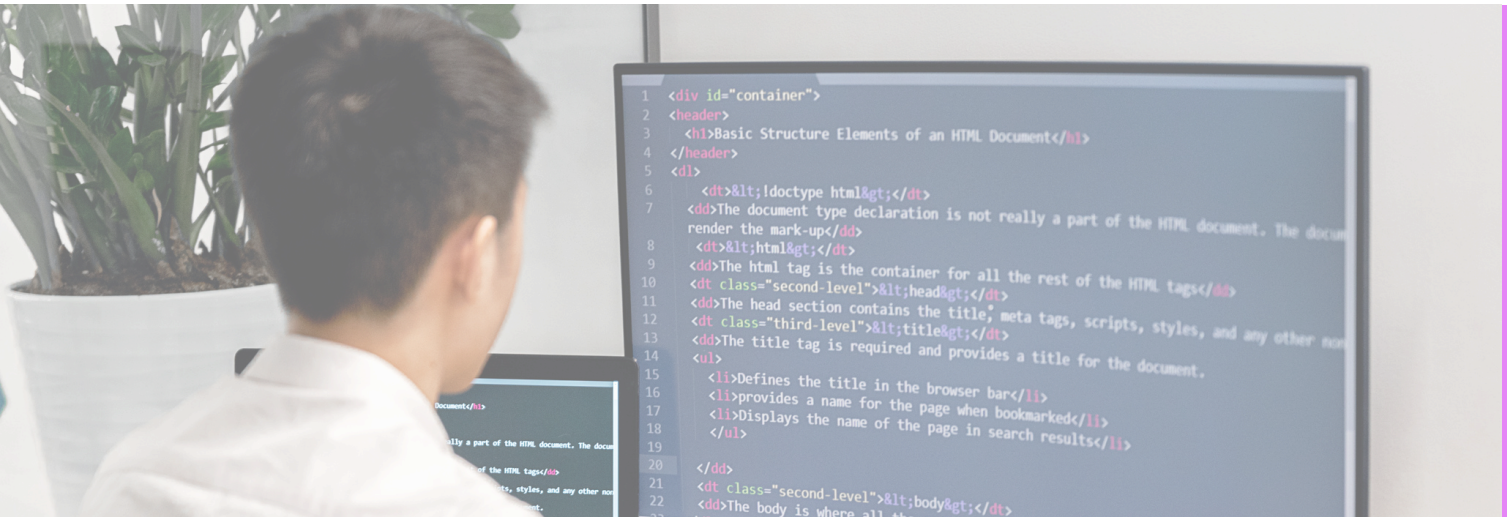
Convergencia y Complementariedad

A pesar de las diferencias, los RNF y los fundamentos de calidad en DevOps no son enfoques excluyentes, sino complementarios Por ejemplo:

Planificación y Ejecución:
Los RNF proporcionan una **base sólida durante la fase de diseño**, mientras que el enfoque DevOps permite **validar y ajustar estos criterios en tiempo real durante la operación** del software.

Herramientas y Prácticas:
La integración de herramientas de automatización, pruebas continuas y monitoreo en producción **facilita la verificación constante de los RNF**, garantizando que los **parámetros definidos inicialmente se cumplan de forma consistente**.

Casos de Éxito en la Industria:
Organizaciones que han logrado integrar un enfoque estructurado de **RNF con prácticas DevOps han alcanzado altos niveles de calidad, resiliencia y satisfacción del cliente**, evidenciando la sinergia entre ambos enfoques.



Impacto de la Falta de Comprensión de la Evolución en el Trabajo en Informática

La **falta de entendimiento** de la evolución de los requerimientos de calidad y la transición hacia DevOps ha generado diversos problemas en el ámbito de la informática, tanto en la gestión de proyectos como en la operación de sistemas en producción. Algunos de los impactos y problemas comunes son:

Problemas de Comunicación y Silos Organizacionales
En The Phoenix Project, se ilustra cómo la falta de colaboración entre equipos de desarrollo, operaciones y otros departamentos conduce a **cuellos de botella y a la ineficiencia en la resolución de problemas**.

Resistencia al Cambio y Rigidez en Procesos
El libro destaca la **resistencia a adoptar nuevas prácticas** debido a estructuras organizativas tradicionales.

Incapacidad para Responder Rápidamente a Incidentes
La evolución hacia un modelo basado en DevOps facilita la **detección y resolución** de problemas en tiempo real. Sin embargo, la **falta de comprensión** de esta transición **impide la implementación de sistemas de monitoreo efectivos**.

Falta de Visión Integral del Ciclo de Vida del Software
No comprender la evolución entre los RNF y DevOps puede llevar a una **visión fragmentada del desarrollo y la operación**, **donde se ignoran las oportunidades de mejora continua**.

Conclusión y Futuro

La transformación de los requerimientos no funcionales en el contexto de DevOps representa una **evolución natural** en la búsqueda de la excelencia en la calidad del software. Mientras que los RNF ofrecen un **marco de referencia estructurado**, la adopción de prácticas DevOps permite implementar una cultura de mejora continua basada en el monitoreo y la adaptación en tiempo real.

La falta de comprensión de esta evolución ha demostrado ser una fuente de problemas comunes en el trabajo en informática, evidenciados a través de ejemplos de The Phoenix Project, como la **falta de comunicación, resistencia al cambio, problemas en la respuesta a incidentes y una visión fragmentada del ciclo de vida del software**. Superar estos desafíos implica la integración de ambos enfoques para alcanzar un equilibrio entre la planificación rigurosa y la flexibilidad operativa.

El futuro del desarrollo de software se orienta **hacia modelos híbridos en los que la previsión tradicional y la agilidad operativa coexistan**, permitiendo a las organizaciones responder eficazmente a las demandas del mercado y mejorar la calidad y resiliencia de sus sistemas.

La continua evolución invita a futuras investigaciones y a la exploración de nuevas metodologías que integren de manera efectiva estos paradigmas donde **el impacto en el costo del software debería ser mitigado con experiencia y monitoreo de las acciones de los involucrados en torno a la calidad esperada**.