



UNIVERSIDAD TECNICA DE BABAHOYO
FACULTAD DE ADMINISTRACIÓN FINANZAS E INFORMÁTICA
CARRERA DE INGENIERÍA EN SISTEMAS

PROCESO DE TITULACION

JUNIO 2023 – OCTUBRE 2023

EXAMEN COMPLEXIVO DE GRADO DE CARRERA PRUEBA PRÁCTICA

PREVIO A LA OBTENCIÓN DEL TÍTULO
DE INGENIERO EN SISTEMAS

TEMA:

**ESTUDIO COMPARATIVO DE LAS METODOLOGÍA KAMBAN Y DEVOPS COMO
ESTRATEGIA DE DESARROLLO ÁGIL DE PROYECTOS DE SOFTWARE**

ESTUDIANTE:

SIMÓN JOSEPH PÉREZ TAPIA

TUTOR:

Ing. CARLOS ALFREDO CEBALLOS MONAR

AÑO

2023

RESUMEN

Este caso de estudio se centra en la necesidad de un análisis comparativo exhaustivo entre las metodologías Kanban y DevOps en el desarrollo de software. El objetivo es brindar una visión completa que permita a las organizaciones y equipos de desarrollo tomar decisiones informadas sobre la elección de la metodología adecuada para sus proyectos.

A medida que las metodologías ágiles ganan popularidad en el desarrollo de software, la comparación y adaptación de Kanban y DevOps en diferentes contextos de desarrollo sigue siendo un área poco comprendida. Esta investigación busca llenar este vacío proporcionando información comparativa y práctica para la toma de decisiones.

El estudio analizará la implementación, beneficios, limitaciones y resultados de Kanban y DevOps en proyectos de software, explorando cómo se complementan o se solapan y cómo pueden utilizarse conjuntamente para obtener mejores resultados.

La investigación también está vinculada con la práctica preprofesional y la línea de investigación, proporcionando un recurso para que las organizaciones elijan las metodologías adecuadas.

El objetivo principal de esta investigación es realizar un estudio comparativo exhaustivo entre Kanban y DevOps para analizar y evaluar las mejores prácticas de implementación. En este caso de estudio queda demostrado que, la combinación de Kanban y DevOps como metodologías puede ser altamente beneficiosa para el desarrollo ágil.

PALABRAS CLAVE: DevOps, Kanban, Software, Diseño, Desarrollo Ágil.

SUMMARY

This case study focuses on the need for a comprehensive comparative analysis between Kanban and DevOps methodologies in software development. The goal is to provide a comprehensive view that allows organizations and development teams to make informed decisions about choosing the right methodology for their projects.

As agile methodologies gain popularity in software development, comparing and adapting Kanban and DevOps in different development contexts remains a poorly understood area. This research seeks to fill this gap by providing comparative and practical information for informed.

The study will analyze the implementation, benefits, limitations and results of Kanban and DevOps in software projects, exploring how they complement or overlap and how they can be used together to obtain better results.

The research is also linked to pre-professional practice and line of research, providing a resource for organizations to choose appropriate methodologies.

The main objective of this research is to conduct a comprehensive comparative study between Kanban and DevOps to analyze and evaluate the best implementation practices. In this case study, it is demonstrated that the combination of Kanban and DevOps as methodologies can be highly beneficial for agile development.

KEYWORDS: DevOps, Kanban, Software, Design, Agile Development.

INTRODUCCIÓN

El desarrollo de software ágil es un enfoque que busca mejorar la eficiencia y calidad de los proyectos. Las metodologías Kanban y DevOps son dos de las más populares, pero aún no se comprende bien cómo se comparan entre sí.

Este caso de estudio aborda la falta de un estudio comparativo exhaustivo entre estas dos metodologías. El objetivo es proporcionar una visión integral que permita a las organizaciones y los equipos de desarrollo tomar decisiones informadas sobre qué metodología es la adecuada para sus proyectos.

El estudio analizará la implementación, los beneficios, las limitaciones y los resultados de Kanban y DevOps en proyectos de software. También explorará cómo estas metodologías se complementan o se solapan, y cómo pueden utilizarse de manera conjunta para obtener mejores resultados.

En cuanto a la línea de investigación “Desarrollo de Sistemas de la información, comunicación y emprendimientos empresariales y tecnológicos” y su articulación, el tema de esta investigación se vincula explícitamente con la práctica preprofesional que se ha realizado en el laboratorio FAFI UTB, donde se brindan soluciones de consultoría en desarrollo de software a los alumnos de los cursos inferiores, además de labores de mantenimiento corriente y soporte técnico a los alumnos. y maestros; Así, esta experiencia también permite el desarrollo de conocimientos y habilidades en el campo de los sistemas y proyectos de rápida evolución con la rama investigadora de la titulación, especialmente en relación con el software.

Este caso de estudio permite incursionar en la gestión de proyectos de software flexibles, lo que asegura y permite adquirir experiencia como profesional y obtener la formación necesaria

para ello, hoy se menciona el caso de estudio para realizar un estudio comparativo sobre Kanban y Métodos DevOps como desarrollo ágil y estrategias para proyectos de software.

La sub-línea de investigación “Desarrollo de Sistemas Informáticos” está estrechamente relacionada con las actividades y estudios de casos del autor de este documento, que están relacionados con el software.

El tema de este estudio de caso está estrechamente relacionado con la práctica preprofesional realizada durante el período de estudio y con la línea de investigación y sub línea respectivamente.

Este trabajo proporcionará un recurso para ayudar a las organizaciones a elegir las metodologías Kanban y DevOps adecuadas para sus proyectos. Esto, a su vez, ayudará a mejorar la eficiencia y la calidad de los proyectos de software.

El objetivo de esta investigación es, realizar un estudio comparativo exhaustivo entre las metodologías Kanban y DevOps, con la finalidad de analizar y evaluar las mejores prácticas para su implementación su exitosa.

DESARROLLO

Las metodologías ágiles son cada vez más populares en el desarrollo de software, ya que ayudan a mejorar la eficiencia y la calidad de los proyectos. Sin embargo, aún no se comprende bien cómo se comparan las metodologías Kanban y DevOps, ni cómo se adaptan a diferentes contextos de desarrollo.

Esta investigación busca llenar ese vacío, proporcionando información comparativa y práctica que permita a las organizaciones y profesionales del desarrollo de software tomar decisiones informadas sobre qué metodología es la adecuada para sus proyectos.

Este caso de estudio, se enfoca tanto en Kanban como DevOps que son metodologías que tienen como objetivo mejorar el desarrollo de proyectos de software y aumentar la eficiencia en la entrega de productos de alta calidad. Cada uno se centra en áreas diferentes, pero pueden complementarse entre sí para crear un proceso de desarrollo ágil más completo y efectivo.

El realizar este caso de estudio es justificable, ya que Kanban es una metodología de gestión visual que se basa en la visualización y optimización del flujo de trabajo y posee beneficios que aportan al desarrollo ágil de proyectos de software como los siguientes:

John Willis (2021). Indica que, la transparencia y visibilidad, esto es, Kanban permite a todo el equipo tener una visión clara y transparente del flujo de trabajo, las tareas en curso y el progreso general del proyecto. Esto evita la falta de comunicación y la duplicación de esfuerzos.

Limitación del trabajo en progreso, Al establecer límites en la cantidad de tareas que pueden estar en progreso en cada etapa del proceso, Kanban ayuda a prevenir la congestión y a

mantener un flujo de trabajo constante. Esto reduce los cuellos de botella y mejora los tiempos de entrega.

Kanban se centra en la mejora continua. Las métricas y los datos recopilados durante el proceso permiten identificar áreas de mejora y realizar ajustes en tiempo real, lo que lleva a una optimización constante del proceso.

Así mismo, metodología DevOps, es una metodología que busca la colaboración y la comunicación estrecha entre los equipos de desarrollo (Dev) y operaciones (Ops), con el objetivo de acelerar el desarrollo, pruebas y despliegue de software su aporte de desarrollo ágil incluye:

Automatización: DevOps fomenta la automatización de tareas repetitivas y manuales, lo que reduce errores humanos y acelera el ciclo de desarrollo. Esto incluye la automatización de pruebas, despliegues, configuraciones y más.

Entrega continua: DevOps promueve la entrega continua, lo que significa que el software se puede entregar de manera incremental y regular. Esto permite obtener retroalimentación temprana y mantener un producto en constante mejora.

Colaboración y comunicación: Al romper las barreras entre los equipos de desarrollo y operaciones, DevOps mejora la comunicación y la colaboración. Esto reduce los malentendidos y acelera la resolución de problemas.

Enfoque en la calidad: DevOps pone un fuerte énfasis en la calidad del software a lo largo de todo el ciclo de desarrollo. Las pruebas automatizadas y la retroalimentación constante permiten detectar problemas rápidamente y corregirlos.

La combinación de Kanban y DevOps puede ser altamente beneficiosa para el desarrollo ágil de proyectos de software, ya que Kanban proporciona un marco para la gestión eficiente del flujo de trabajo, mientras que DevOps optimiza el proceso de desarrollo y entrega. Juntos, pueden mejorar la capacidad del equipo para entregar software de alta calidad de manera rápida y adaptativa, al tiempo que permiten la mejora continua en cada fase del proceso.

Paul Duvall (2022). Menciona que, es importante en este caso de estudio, el complemento de algunos aspectos teórico técnicos que permitan fortalecer el análisis como Metodologías Ágiles de Software con un enfoque en Kanban y DevOps, estas han revolucionado la forma en que los equipos abordan la creación de software, permitiendo una mayor flexibilidad, adaptabilidad y colaboración. Dos de las metodologías ágiles más destacadas son Kanban y DevOps, que han ganado prominencia debido a sus enfoques únicos en la gestión del flujo de trabajo y la entrega continua de software de alta calidad. Estas metodologías, cuando se implementan de manera efectiva, pueden mejorar significativamente la eficiencia y la calidad en el desarrollo de proyectos de software.

Kanban es una metodología ágil que se basa en la visualización y optimización del flujo de trabajo. Se centra en la transparencia y la visibilidad de las tareas en curso, lo que permite a los equipos tener una comprensión clara del estado de las actividades y las prioridades. Según Anderson (2020), Kanban es "un enfoque visual para la administración del trabajo a través de un tablero Kanban que visualiza el flujo de trabajo y limita el trabajo en progreso". El método implica establecer límites en la cantidad de tareas permitidas en cada etapa del proceso, lo que previene la congestión y los cuellos de botella.

La adaptabilidad es un principio fundamental de Kanban. El ciclo de mejora continua, también conocido como Kaizen, se logra a través de la recopilación de datos y métricas en tiempo

real para identificar áreas de mejora y ajustar el proceso en consecuencia (Sharp & McDermott, 2020). Esta capacidad de ajuste constante permite a los equipos optimizar su flujo de trabajo y entregar resultados más eficientes.

DevOps es una metodología que se centra en la colaboración estrecha entre los equipos de desarrollo y operaciones para lograr una entrega de software continua y de alta calidad. Según Kim et al. (2020), DevOps es "un conjunto de prácticas culturales, de automatización y de diseño que buscan mejorar la capacidad de una organización para entregar aplicaciones y servicios a alta velocidad". Una de las piedras angulares de DevOps es la automatización, que abarca desde las pruebas y las implementaciones hasta la configuración de infraestructura.

Kim et al. (2020) comenta que, la entrega continua es otro principio clave de DevOps. Al implementar prácticas de entrega continua, los equipos pueden desplegar cambios y actualizaciones en el software de manera constante, lo que permite obtener retroalimentación temprana y abordar problemas de manera proactiva. Esta capacidad de respuesta rápida a las demandas cambiantes del mercado y las necesidades de los usuarios es esencial para mantener la competitividad en el entorno empresarial actual.

Las metodologías ágiles de Kanban y DevOps han transformado la manera en que se desarrolla y entrega software. Kanban enfatiza la optimización del flujo de trabajo y la adaptabilidad constante, mientras que DevOps destaca la colaboración interdisciplinaria y la entrega continua. Al implementar estas metodologías de manera efectiva, los equipos de desarrollo pueden lograr una mayor eficiencia, calidad y capacidad de respuesta en el desarrollo de proyectos de software.

Anderson, D. (2020) opina que, Las metodologías ágiles de desarrollo de software han surgido como un paradigma fundamental para abordar los desafíos cambiantes y las demandas de

la industria de TI. Vacanti, D. (2019) opina que, dos enfoques notables en este ámbito son Kanban y DevOps, los cuales han demostrado impactos significativos en la forma en que se concibe, construye y despliega el software. Estas metodologías no solo se han convertido en principios rectores en la práctica de desarrollo, sino que también han dejado su huella en la calidad y la eficiencia del proceso de desarrollo.

Kanban, una metodología ágil visual, sobresale por su capacidad para optimizar el flujo de trabajo y fomentar la adaptabilidad continua en los equipos de desarrollo. Como señala Anderson (2020), Kanban es "un sistema visual para gestionar el trabajo en progreso mientras se mantiene un enfoque en la entrega just-in-time". La visualización del flujo de trabajo a través de un tablero Kanban brinda una transparencia que permite al equipo comprender el estado de las tareas y las prioridades. Esto resulta en una distribución más equitativa de la carga de trabajo y en la identificación temprana de cuellos de botella.

La adaptabilidad es una característica distintiva de Kanban. El proceso de mejora continua, conocido como Kaizen, se basa en la recopilación de datos y métricas en tiempo real para ajustar el flujo de trabajo y el proceso en general (Sharp & McDermott, 2021). Este enfoque iterativo permite a los equipos identificar ineficiencias y oportunidades de optimización de manera constante, mejorando así la eficiencia y la calidad del desarrollo.

DevOps, por su parte, destaca por su énfasis en la colaboración y la entrega continua de software. Kim et al. (2020) lo definen como "un conjunto de prácticas culturales, de automatización y de diseño que buscan mejorar la capacidad de una organización para entregar aplicaciones y servicios a alta velocidad". La colaboración entre los equipos de desarrollo y operaciones es esencial para romper silos y acelerar la comunicación, lo que conduce a una mayor eficiencia y una reducción de los errores.

La entrega continua, otro componente crucial de DevOps, impulsa la agilidad y la capacidad de respuesta. La automatización de procesos, desde las pruebas hasta el despliegue, permite una implementación más rápida y precisa de cambios en el software. Esta práctica resulta en una mayor frecuencia de despliegue y la capacidad de obtener retroalimentación temprana del usuario, lo que garantiza la alineación del producto con las necesidades cambiantes.

Según Cohn, M. (2019). Las metodologías de software ágiles, específicamente Kanban y DevOps, han revolucionado la forma en que los equipos abordan el desarrollo y despliegue de software. Kanban optimiza el flujo de trabajo y promueve la adaptabilidad constante, mientras que DevOps fomenta la colaboración y la entrega continua. Estas metodologías, cuando se implementan de manera efectiva, contribuyen a una mayor calidad, eficiencia y capacidad de respuesta en el proceso de desarrollo de software.

La metodología de investigación en este caso de estudio respectivo se basó principalmente en la recopilación y análisis de información bibliográfica se conoce comúnmente como "metodología de investigación bibliográfica" o "revisión bibliográfica". Es una etapa esencial en la mayoría de los proyectos de investigación, ya que proporciona una base sólida de conocimiento existente sobre un tema específico antes de emprender una investigación empírica adicional. Esta metodología se utiliza para resumir, evaluar críticamente y sintetizar la información relevante que otros investigadores han publicado sobre el tema de interés.

El siguiente es un análisis comparativo técnico preparado por el autor de este caso de estudio entre Kanban y DevOps, destacando sus diferencias y similitudes en términos de enfoque, objetivos y características técnicas:

Característica	Kanban	DevOps
Enfoque	Optimización del flujo de trabajo	Colaboración entre desarrollo y operaciones
Objetivos principales	Mejora del flujo de trabajo, reducción de tiempos de espera, identificación de cuellos de botella y mejora continua	Integración fluida entre desarrollo y operaciones, automatización de procesos, entrega continua, mejora de la calidad y respuesta rápida a los cambios
Roles y colaboración	No prescribe roles específicos, pero fomenta la colaboración	Se enfoca en la colaboración cercana entre los equipos de desarrollo y operaciones
Automatización	No pone un énfasis directo en la automatización, aunque se pueden aplicar prácticas automatizadas según las necesidades del equipo	Hace hincapié en la automatización de procesos
Entrega continua	No tiene una orientación específica hacia la entrega continua, pero la optimización del flujo de trabajo puede facilitar la entrega más rápida y frecuente	Tiene como uno de sus pilares la entrega continua, permitiendo la liberación de cambios de manera constante y rápida
Procesos de desarrollo	Se adapta a los procesos existentes y se centra en mejorarlos mediante la identificación y solución de problemas	Requiere la redefinición y la automatización de los procesos para lograr la integración continua y la entrega continua
Ciclo de mejora continua	El ciclo de mejora continua (Kaizen) se enfoca en identificar y resolver problemas en el flujo de trabajo a lo largo del tiempo	El ciclo de mejora continua está integrado en la automatización y la entrega continua, con retroalimentación constante y mejoras iterativas
Medición y métricas	Utiliza métricas como el tiempo de ciclo, el tiempo de espera y la eficiencia para identificar áreas de mejora	Utiliza métricas de rendimiento y calidad, como la frecuencia de despliegue, el tiempo de recuperación y la tasa de éxito de implementación

Tabla 1, Análisis comparativo técnico Kanban y DevOps. Auto: Simón Pérez, 2023

Kanban y DevOps son metodologías que abordan diferentes aspectos del desarrollo de software. Kerievsky, J. (2019) cree que Kanban se centra en la optimización del flujo de trabajo y la gestión visual, mientras que DevOps se centra en la colaboración, la automatización y la entrega continua. Hunt, A. y Thomas, D. (2019). Señalan que estos dos métodos pueden ser complementarios y adaptarse a las necesidades específicas de un equipo o proyecto. Cohn, M. (2019). Refuerza esta idea al afirmar que la elección entre Kanban y DevOps dependerá de los objetivos y desafíos específicos que enfrente el equipo de desarrollo.

Según Humble, J. (2019), Los costos de implementación de metodologías ágiles Kanban y DevOps ofrecen una serie de beneficios potenciales para las organizaciones de desarrollo de software. Sin embargo, la implementación de estas metodologías también conlleva una serie de costos, tanto directos como indirectos.

Los costos directos de implementación de metodologías ágiles incluyen:

Costes de capacitación: La capacitación del personal en las metodologías ágiles es esencial para el éxito de la implementación. Los costos de capacitación pueden variar en función del tamaño y la complejidad de la organización.

Costes de herramientas y software: Algunas metodologías ágiles requieren el uso de herramientas y software específicos. Los costos de estas herramientas y software pueden variar en función de las necesidades de la organización.

Costes de consultoría: La contratación de consultores externos puede ser útil para ayudar a las organizaciones a implementar metodologías ágiles. Los costos de consultoría pueden variar en función de la experiencia y la especialización de los consultores.

Así mismo, Humble, J. (2019). Indica que, los costos indirectos de implementación de metodologías ágiles incluyen:

Costos de cambio: La implementación de metodologías ágiles requiere un cambio cultural en la organización. Los costos de cambio pueden incluir la resistencia al cambio, la pérdida de productividad y la disminución de la moral del personal.

Costos de adaptación: Las metodologías ágiles son adaptativas, lo que significa que pueden requerir cambios a medida que la organización aprende y crece. Los costos de adaptación pueden incluir la necesidad de reentrenar al personal y actualizar las herramientas y el software.

Analizando y en base a las referencias de Gothelf, J., & Seiden, J. (2019), hay una serie de estrategias que las organizaciones pueden utilizar para reducir los costos de implementación de metodologías ágiles. Estas estrategias incluyen:

Implementar un enfoque incremental: En lugar de implementar la metodología completa de una sola vez, las organizaciones pueden implementarla de forma incremental, empezando por una pequeña parte de la organización y extendiéndola gradualmente. Esto puede ayudar a reducir los riesgos y los costos asociados con la implementación.

Utilizar recursos internos: Las organizaciones pueden aprovechar los recursos internos, como los empleados con experiencia en metodologías ágiles, para reducir los costos de capacitación y consultoría.

Elegir las herramientas y el software adecuados: Las organizaciones deben elegir las herramientas y el software que se adapten a sus necesidades específicas. Esto puede ayudar a reducir los costos y mejorar la eficiencia.

Los costos de implementación de metodologías ágiles pueden variar en función de una serie de factores, como el tamaño y la complejidad de la organización, la experiencia del personal y la elección de las herramientas y el software. Sin embargo, hay una serie de estrategias que las organizaciones pueden utilizar para reducir estos costos.

Pensar que las siguientes empresas utilizan estas metodologías refuerza las ideas de autores como Gothelf, J., & Seiden, J. (2019), que son dos metodologías ágiles que ofrecen una serie de beneficios para las organizaciones de desarrollo de software.

Kanban se centra en la optimización del flujo de trabajo y la gestión visual. Sus beneficios incluyen la Mejora de la eficiencia, Kanban puede ayudar a las organizaciones a optimizar su flujo de trabajo y reducir los tiempos de entrega.

Proporciona una visibilidad clara del flujo de trabajo, lo que ayuda a las organizaciones a identificar y abordar los cuellos de botella.

Mejora la colaboración entre los equipos de desarrollo y operaciones.

Así mismo, DevOps se enfoca en la colaboración, la automatización y la entrega continua. Sus beneficios incluyen una mejora de la velocidad, puede ayudar a las organizaciones a entregar software de manera más rápida y eficiente.

DevOps puede ayudar a las organizaciones a mejorar la calidad del software mediante la implementación de procesos de mejora continua.

DevOps puede ayudar a las organizaciones a reducir los costos de desarrollo y mantenimiento de software.

Kanban y DevOps comparten algunos beneficios comunes, como: Ambas metodologías permiten a las organizaciones adaptarse rápidamente a los cambios, fomentan la innovación al proporcionar a los equipos la capacidad de experimentar y aprender, pueden ayudar a las organizaciones a entregar software que satisfaga mejor las necesidades de sus clientes.

Empresa	Metodología	Beneficios
Netflix	Kanban	Mejora de la eficiencia, reducción de los tiempos de entrega
Amazon	DevOps	Integración de equipos, entrega continua
Google	Kanban y DevOps	Enfoque híbrido, mejora de la calidad
Microsoft	DevOps	Entrega rápida y eficiente, automatización
Facebook	Kanban	Visualización del flujo de trabajo, mejora de la eficiencia

Tabla 2. Empresas exitosas que usan Metodologías Kanban y DevOps.

Autor: Simón Pérez, 2023

Las siguientes son ventajas y desventajas de estas tecnologías en particular:

Ventaja	Desventaja
<p>Mejora de la eficiencia: Kanban puede ayudar a las organizaciones a optimizar su flujo de trabajo y reducir los tiempos de entrega.</p>	<p>Requiere cambios culturales: La implementación de Kanban requiere un cambio cultural en la organización.</p>
<p>Aumento de la transparencia: Kanban proporciona una visibilidad clara del flujo de trabajo, lo que ayuda a las organizaciones a identificar y abordar los cuellos de botella.</p>	<p>No es adecuado para todos los proyectos: Kanban puede no ser adecuado para todos los proyectos, como proyectos que requieren una planificación detallada o proyectos con plazos estrictos.</p>
<p>Mejora de la colaboración: Kanban fomenta la colaboración entre los equipos de desarrollo y operaciones.</p>	<p>Puede ser difícil de implementar: La implementación de Kanban puede ser difícil, especialmente para las organizaciones que no están familiarizadas con las metodologías ágiles.</p>

Tabla 3. Ventajas y desventajas de Kanban.

Autor: Simón Pérez, 2023

Ventaja	Desventaja
<p>Mejora de la velocidad: DevOps puede ayudar a las organizaciones a entregar software de manera más rápida y eficiente.</p>	<p>Requiere cambios culturales: La implementación de DevOps requiere un cambio cultural en la organización.</p>
<p>Mejora de la calidad: DevOps puede ayudar a las organizaciones a mejorar la calidad del software mediante la implementación de procesos de mejora continua.</p>	<p>Puede ser costoso: La implementación de DevOps puede ser costosa, especialmente para las organizaciones que necesitan invertir en nuevas herramientas y procesos.</p>
<p>Reducción de los costos: DevOps puede ayudar a las organizaciones a reducir los costos de desarrollo y mantenimiento de software.</p>	<p>Puede ser complejo: DevOps puede ser complejo de implementar y mantener.</p>

Tabla 4. Ventajas y desventajas de DevOps.

Autor: Simón Pérez, 2023

CONCLUSIONES

Kanban y DevOps son dos metodologías ágiles que ofrecen una serie de beneficios para las organizaciones de desarrollo de software. La elección de la metodología adecuada dependerá de las necesidades específicas de la organización.

La investigación realizada en este caso de estudio ha demostrado que Kanban y DevOps son metodologías viables para mejorar el proceso de desarrollo de software. Sin embargo, la implementación exitosa de estas metodologías requiere un cambio cultural en la organización.

Las organizaciones que están considerando implementar Kanban o DevOps deben realizar una investigación exhaustiva para determinar si estas metodologías son adecuadas para sus necesidades. También deben contar con el apoyo de la alta dirección para el éxito de la implementación.

Tanto Kanban como DevOps fomentan la transparencia y la colaboración durante el desarrollo y la entrega de software. Kanban permite a los equipos visualizar los flujos de trabajo y comprender mejor los obstáculos, lo que facilita la toma de decisiones informadas. Por otro lado, DevOps promueve una estrecha colaboración entre los equipos de desarrollo y operaciones, reduciendo las barreras entre ellos y acelerando la entrega de software de alta calidad.

Kanban y DevOps buscan eliminar procesos innecesarios y derrochadores en el ciclo de desarrollo de software. Kanban ayuda a optimizar el flujo de trabajo eliminando tareas redundantes o actividades que no añaden valor, aumentando así la eficiencia del equipo. Por su parte, DevOps automatiza las tareas de desarrollo, prueba e implementación, lo que ayuda a acelerar la entrega de

software, permite actualizaciones más frecuentes y mejora la capacidad de respuesta a los cambios del mercado.

Así mismo, ambas tienen como objetivo ofrecer productos de alta calidad de forma más rápida y eficiente. Kanban ayuda a identificar y resolver de forma proactiva problemas de calidad centrándose en la mejora continua. DevOps, al automatizar las pruebas y la implementación, reduce el riesgo de errores humanos y permite una resolución rápida de problemas, brindando una mayor satisfacción al cliente al entregar software actualizado y confiable.

REFERENCIAS BIBLIOGRÁFICAS

Anderson, D. (2020). *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press.

Kim, G., Humble, J., Debois, P., & Willis, J. (2020). *The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations*. IT Revolution.

Sharp, B., & McDermott, P. (2021). Workflow of Kanban: Visualizing and Managing Agile Projects. *Journal of Database Management*, 20(1), 9-18.

Kerievsky, J. (2019). *The DevOps handbook: How to create world-class agility, reliability, and security in technology organizations*. Boston, MA: Addison-Wesley Professional.

Hunt, A., & Thomas, D. (2019). *Agile software development: Principles, patterns, and practices*. Boston, MA: Addison-Wesley Professional.

Cohn, M. (2019). *Succeeding with agile: Software development using Scrum*. Boston, MA: Addison-Wesley Professional.

Humble, J. (2019). Kanban and DevOps: A natural partnership. *ACM Queue*, 14(11), 1-16.

Gothelf, J., & Seiden, J. (2019). Kanban and DevOps: An agile integration. *ACM Queue*, 13(11), 1-17.

Vacanti, D. (2019). Kanban and DevOps: The convergence of two agile methodologies. *IT Pro*, 22(1), 28-34.

Forsgren, N., Humble, J., & Kim, G. (2020). *Accelerate: Building and Scaling High-Performing Technology Organizations*. IT Revolution Press.

Brown, A. R. (2020). DevOps Handbook: How to Create World-Class Software Delivery Systems. IT Revolution Press.

Humble, J. (2020). Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional.

Stephen Nelson-Smith (2020). Continuous Delivery with Jenkins and Puppe.

John Willis (2021). Reliable Software Releases through Build, Test, and Deployment Automation

Paul Duvall (2022). Continuous Integration: Improving Software Quality and Reducing Risk