

DevOps y la experiencia del desarrollador

DevOps: Acelerando el ciclo de desarrollo al reducir la fricción

Comencemos aprendiendo los fundamentos de DevOps.

El mundo de la tecnología tiene varios términos. A veces, tales términos pueden ser abstractos, y las personas tienden a aplicar varias interpretaciones a un escenario específico. DevOps no es una excepción. Además, al referirse a DevOps, considerar perspectivas desde un individuo, un equipo y toda la organización puede, a veces, hacer que no quede claro cómo interactúan varios componentes entre sí.

Antecedentes de DevOps

En el pasado, el software era esencialmente algo que instalabas. Entonces, para un equipo de desarrollo de productos, todo lo que necesitabas eran ingenieros que pudieran desarrollarlo y probarlo. Hablando de pruebas, todo se trataba de aplastar errores antes del lanzamiento. Solo imagínalo por un momento; era similar al lanzamiento de los primeros videojuegos, muy parecido a la era del Nintendo Entertainment System (NES). Una vez que un juego se enviaba en un cartucho, no había oportunidad de actualizaciones ni de agregar nuevas características. Los errores que se enviaban con el juego permanecían con el consumidor para siempre. Sin embargo, con la aparición del Servidor HTTP de Apache y el cambio hacia la comunicación en red, surgió la necesidad de que las operaciones de IT gestionaran este nuevo paradigma.

En muchos casos, el equipo de operaciones de IT se especializaba en operaciones de IT. Sus herramientas, cultura e incluso sus objetivos eran diferentes de los del equipo de desarrollo. El equipo de desarrollo quería agregar características, mientras que el equipo de IT consideraba que la estabilidad del sistema era su máxima prioridad. Esto era una contradicción.

Además, en el mundo dividido de estos dos equipos, cuando se trataba de la implementación, se sentía como lanzar código por encima del muro. Los desarrolladores lanzaban un artefacto perfectamente implementable al equipo de operaciones de IT en el otro lado del muro. Sin embargo, por alguna razón, el equipo de operaciones de IT recibía algo incompleto y no funcional para el entorno de producción, lo que llevaba a la extenuación mientras luchaban por implementarlo y mantenerlo de alguna manera:



En los últimos años, ha habido un notable avance tecnológico, y su influencia ha permeado mucho más allá del ámbito del software, tocando prácticamente todas las industrias. ¿Por qué es esto? Esto se debe a que el desarrollo de software juega un papel crucial en las estrategias de las organizaciones, independientemente de su tamaño, en la gestión empresarial moderna. Las empresas afectadas por esta ola tecnológica se encuentran en un panorama competitivo donde el enfoque está en ofrecer productos rápidos, vanguardistas y confiables a sus clientes.

A pesar de la posibilidad de cambios significativos en las condiciones del mercado y las necesidades del cliente en menos de un año, lanzar software solo unas pocas veces al año no es suficiente. No podemos permitir retrasos en los lanzamientos de software debido a la fricción entre silos internos. Un silo denota un sistema, proceso o departamento aislado que carece de integración o comunicación con otras partes, lo que puede causar ineficiencias. Debemos encontrar una manera de agilizar el proceso desde el desarrollo del producto hasta la operación. Aquí es donde entran en juego una metodología completamente nueva de desarrollo, estructura organizativa y cultura.

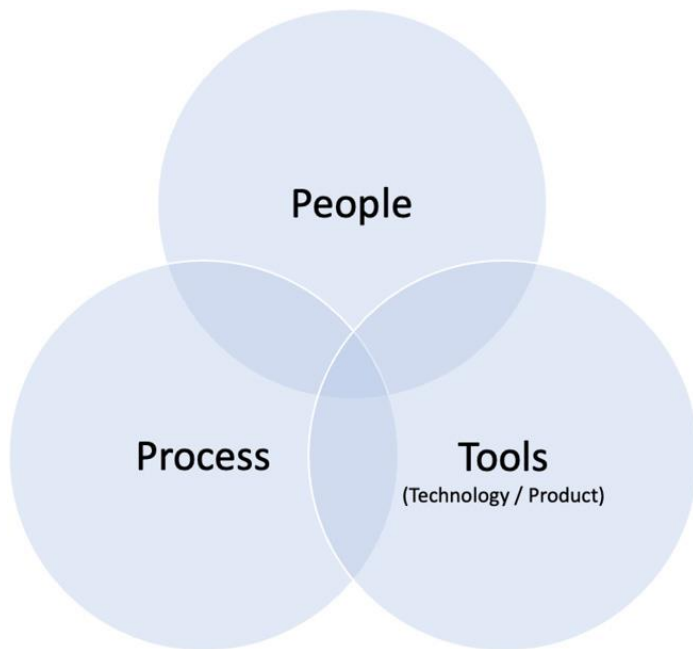
DevOps surge como una metodología nacida de este mismo escenario. Representa una fusión de desarrollo y operaciones, donde los equipos de desarrollo y operaciones colaboran como una entidad unificada para inculcar una cultura compartida, refinar procesos y desplegar herramientas, todo en la búsqueda de acelerar los lanzamientos, mejorar la calidad general del producto, recopilar comentarios de los clientes y, en última instancia, brindar un mejor y más rápido servicio a los clientes. Esto, a su vez, conduce a un desempeño más efectivo en el mercado.

¿Qué es DevOps?

El término DevOps es una combinación de las palabras development (desarrollo) y operations (operaciones).

En esencia, DevOps trasciende la simple integración de roles o equipos; representa un cambio cultural más amplio dentro de la organización. Se trata de alinear prioridades, simplificar flujos de trabajo y, en última instancia, dismantlar silos que obstaculizan la comunicación y el progreso.

efectivos. En resumen, DevOps es la unión de personas, procesos y herramientas para proporcionar continuamente valor a los usuarios finales y es, fundamentalmente, un cambio cultural:



Importante: Muchas personas piensan que lo entienden aquí y pasan a implementar un pipeline de automatización para DevOps. Pero espera. Simplemente reemplazar estos elementos no garantizará el éxito de una iniciativa DevOps. Para que DevOps sea exitoso, necesitas saber más sobre lo que es, lo que no es y por qué lo es.

En realidad, la definición de DevOps es bastante amplia. En consecuencia, el concepto de Dev*Ops, una forma extendida de DevOps, ha estado expandiéndose rápidamente recientemente. Se han propuesto varios enfoques derivados, como DevSecOps y BizDevOps, lo que ha llevado a una multitud de perspectivas sobre lo que abarca DevOps.

Desafortunadamente, no existe una definición universalmente aceptada y precisa de DevOps, y la forma de hacer algo llamado DevOps es respondida de manera diferente por diferentes personas. Como resultado, los malentendidos e interpretaciones incorrectas sobre DevOps son comunes, lo que a menudo lleva a disputas. Y a veces, las personas no quieren hablar de ello cuando llega el momento de hablar sobre hacer DevOps, ¡aunque sea tan importante!.

Qué NO es DevOps?

A menudo, al definir lo que es DevOps, puede ser más fácil indicar lo que NO es DevOps en lugar de tratar de especificar lo que realmente es DevOps. Aclaremos un malentendido típico aquí.

DevOps NO se trata solo de una herramienta, tecnología o producto

Existen numerosas herramientas disponibles en el mercado que pueden aplicarse a DevOps. Muchos proveedores de nube y herramientas las promocionan diciendo: “Con esta herramienta, puedes construir un flujo de trabajo DevOps”, o “Esta herramienta es esencial para un equipo DevOps.”

Cuando muchas personas escuchan el término DevOps, algunas de ellas piensan en tecnologías de nube como Amazon Web Services (AWS), Azure y Google Cloud Platform (GCP), o en tecnologías

como Docker y Kubernetes. Además, el ecosistema de Kubernetes también incluye muchos componentes como Istio, Flux, Helm, Envoy y Prometheus. Hay varias plataformas, incluyendo GitHub Actions, CircleCI y Jenkins, que facilitan lanzamientos continuos, rápidos y frecuentes. El mundo de las herramientas de monitoreo es igualmente diverso. Si aún no estás familiarizado con estos nombres, podría llevar un tiempo entender sus matices y beneficios.

Sin embargo, incluso si dominas estas herramientas y sigues arquitecturas conocidas o historias de éxito, eso no significa que hayas logrado DevOps. Porque siempre existen personas y procesos detrás de la herramienta, y solo cambiar herramientas no los cambiará.

En realidad, no es raro ver un desarrollo al estilo waterfall incluso cuando se utilizan herramientas con la marca DevOps. Respetar las reglas complejas existentes de la empresa insertando procesos de aprobación de múltiples pasos en flujos de trabajo automatizados, restringir las liberaciones a los clientes a una vez cada 3 meses para evitar cambios en la contabilidad o en las verificaciones de seguridad, o hacer que el equipo de infraestructura gestione los contenedores creados por el equipo de aplicaciones utilizando una plataforma construida con el moderno sistema de orquestación, Kubernetes: hacer todas estas cosas solo aumentará la carga operativa y la confusión en lugar de crear valor para el negocio

En última instancia, lograr objetivos comerciales significativos y transformaciones requiere cambios en las personas, las organizaciones y los procesos. Entonces, ¿se trata DevOps de transformar individuos y organizaciones?

DevOps NO se trata de individuos específicos, equipos o roles

DevOps es una realidad, pero el término en sí mismo es algo así como una palabra de moda. Las empresas contratan ingenieros para estas nuevas actividades con títulos como Ingeniero DevOps o Arquitecto DevOps, e incluso se refieren a sus equipos como equipos DevOps. Las personas que piensan que no pueden hacer DevOps por sí mismas pueden incluso cometer el error de intentar subcontratar DevOps a socios de DevOps.

Cuando observas esta situación, podría parecer que DevOps se refiere a individuos específicos, equipos o roles, pero ese no es el caso. En muchos casos, estos términos simplemente se refieren a roles como super ingenieros de infraestructura, super desarrolladores o simplemente ingenieros de la nube.

Las empresas hacen esto porque, para responder rápidamente a las necesidades comerciales en evolución y lograr liberaciones rápidas, requieren nuevas tecnologías y automatización. Además, a menudo tienen que gestionar plataformas complejas que incluyen tanto componentes existentes como nuevos. Esto incluye hacer el máximo uso de GitHub Actions, que se trata en este curso. **Áreas que solían involucrar instalaciones manuales en el dominio de la infraestructura, configuraciones a través de pantallas o operaciones CLI, ahora requieren estar bajo control de versiones con Git y la configuración y gestión de flujos de trabajo automatizados.**

Sin embargo, en la práctica, DevOps trata con asuntos aún más complejos. No se trata solo de dominar estas herramientas y tecnologías o de configurar roles para ellas. En realidad, requiere líderes fuertes que puedan liderar la transformación dentro de los sistemas y organizaciones existentes. A menudo, se involucran en acciones que van más allá del marco amplio de DevOps.

Muchas de las personas que he visto que realmente han logrado DevOps están persiguiendo objetivos más grandes que solo la implementación de DevOps.

DevOps es esencialmente un viaje para transformar organizaciones que manejan una variedad de tecnologías y productos complejos, y no se trata simplemente de agregar personas con nuevas habilidades a los equipos existentes o de cambiar las definiciones de equipos o trabajos.

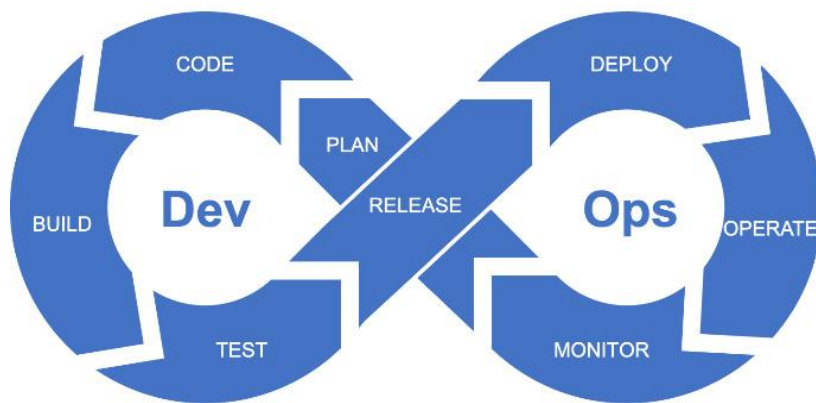
DevOps NO es solo un proceso

Los equipos DevOps a menudo se involucran en un desarrollo iterativo en ciclos cortos de 2-3 semanas, comúnmente llamados sprints, y realizan reuniones stand-up todas las mañanas. Esto es cierto, y al adoptar DevOps, puedes incorporar procesos que se adhieren a las mejores prácticas del proceso Ágil y Scrum. Estas son metodologías en el desarrollo de software que dividen cada funcionalidad en partes más pequeñas y llevan el desarrollo a través de numerosos ciclos cortos.

Entonces, ¿DevOps es un concepto evolucionado de Ágil o Scrum, o los abarca? La respuesta a esta pregunta es tanto sí como no.

DevOps amplía el alcance para no solo enfocarse en el desarrollo de software como Ágil, sino también en el lanzamiento del software, la recopilación de comentarios y las mejoras. Estos principios y filosofías se aplican a lo largo de todo el ciclo de vida del desarrollo de software (SDLC). Los equipos se centran en todo el ciclo de vida del producto, no solo en desarrollar nuevas funciones o diseñar componentes web.

La figura muestra las ocho fases de un ciclo de devops:



Sin embargo, el objetivo final de DevOps no es introducir un nuevo proceso en sí mismo.

DevOps tiene una amplia gama de prácticas. Por ejemplo, un entorno que necesita implementar aplicaciones en numerosos servidores, como Microsoft, es diferente de una situación en la que se despliega una pequeña aplicación en una instancia EC2. Las prácticas a aplicar también difieren entre un servicio con una base de usuarios de 1 millón y un servicio con una base de usuarios de 1,000.

Consideremos Ágil, citando el blog de Andy Hunt:

Los métodos ágiles piden a los practicantes que piensen, y francamente, eso es difícil de vender. Es mucho más cómodo simplemente seguir las reglas que se te han dado y afirmar que estás “haciéndolo según el libro”. Es fácil, es seguro de ridículo o recriminación; no te despedirán por ello. Aunque podríamos denunciar públicamente los estrechos confines de un conjunto de reglas, hay seguridad y comodidad allí. Pero, por supuesto, ser ágil o efectivo no se trata de comodidad.

Esta idea también se aplica a DevOps. Para lanzar frecuentemente y rápidamente desarrollos centrados en el cliente, ofrecer un servicio confiable y, en última instancia, hacer un impacto significativo en el negocio, necesitas abordar los problemas desde una perspectiva más amplia que solo herramientas, personas y procesos.

DevOps es cultura

Entonces, DevOps es más que una fusión de desarrollo y operaciones, roles específicos, procesos, herramientas, tecnología o productos. Pero, a menudo, las personas lo simplifican como personas, procesos y herramientas.

Volvamos a las palabras de Patrick Debois, una figura prominente en la comunidad DevOps, para obtener claridad:

Mi definición actual de Dev*Ops: todo lo que haces para superar la fricción creada por los silos... Todo lo demás es ingeniería pura.

En la era actual de la tecnología en la nube, tanto las empresas vanguardistas como las tradicionales, junto con las start-ups, pueden acceder fácilmente a los mismos entornos. Incluso las herramientas impulsadas por IA están disponibles a un costo asequible.

En última instancia, lo que obstaculiza el progreso de tu equipo y negocio es la fricción, y esta fricción a menudo surge dentro de los silos que abarcan personas, procesos y herramientas.

DevOps, en su núcleo, representa un cambio cultural y un enfoque diseñado para eliminar la fricción entre los silos dentro de una organización. Implica un cambio en la mentalidad, los hábitos y la cultura.

Principios de DevOps

Vamos a profundizar en los principios fundamentales de DevOps para obtener una comprensión más completa de lo que realmente implica esta cultura.

Centrado en el cliente

Toda acción en cada proceso debe tener como objetivo proporcionar valor al cliente.

Enfocarse en el cliente al construir y mantener software permite una entrega más rápida y relevante de características. Los ciclos de retroalimentación cortos contribuyen a ajustar el producto en alineación con las necesidades del usuario, reduciendo riesgos y maximizando el valor al menor costo posible. Al centrarse en las necesidades del cliente, el objetivo final se convierte en resolver problemas del mundo real de manera eficiente, haciendo más fácil priorizar tareas, gestionar acumulaciones y agilizar la asignación de recursos, haciendo que las operaciones sean eficientes y rentables. En última instancia, esto sienta las bases para el éxito empresarial a largo plazo al satisfacer mejor las demandas del mercado y las expectativas de los usuarios.

En DevOps, estar centrado en el cliente no es solo una frase de moda; es una necesidad.

Crear con el final en mente

Entender las necesidades del cliente y abordar problemas del mundo real debe tener prioridad sobre operar bajo meras suposiciones. Esto aboga por una estrategia holística en la que los equipos sincronicen tanto tareas de desarrollo como operacionales para satisfacer los requisitos del cliente. Esto incluye una comprensión del ciclo de vida completo del producto, desde su concepción y desarrollo hasta su implementación y soporte continuo, asegurando que el producto final realmente beneficie a los usuarios finales.

Descuidar las necesidades del cliente desde el principio puede resultar en un producto técnicamente impecable que, sin embargo, no cumple con las expectativas del usuario y la capacidad de resolver problemas. Este principio actúa como un recordatorio constante para alinear todos los esfuerzos tecnológicos con los objetivos comerciales y las expectativas de los usuarios, asegurando que el producto final no solo sea funcional, sino también valioso y relevante para la audiencia.

Equipos autónomos y multifuncionales

En el núcleo de la implementación exitosa de DevOps está la estrategia indispensable de formar equipos autónomos y multifuncionales. A diferencia de los entornos de ingeniería tradicionales, donde desarrolladores, operaciones y especialistas en aseguramiento de la calidad (QA) operan en compartimentos separados, DevOps defiende la disolución de estas barreras. El punto clave aquí no es solo la formación de equipos diversos, sino hacer que estos equipos sean autónomos, capaces de supervisar un producto o característica desde su concepción hasta su entrega.

¿Por qué es esto esencial? La respuesta radica en la agilidad y eficiencia. Cuando un equipo posee una variedad de habilidades, desde la codificación y pruebas hasta la implementación y diseño, e incluso una comprensión matizada del negocio, la velocidad de toma de decisiones se acelera. Esto reemplaza la cadena de mando lenta y engorrosa inherente a los sistemas jerárquicos con una cultura de acción rápida y responsable.

La eliminación de estos cuellos de botella organizacionales no solo agiliza el flujo de trabajo, sino que también cultiva una cultura de propiedad y responsabilidad. El equipo no tiene que esperar a departamentos externos o superiores para tomar decisiones; tienen el conjunto de habilidades colectivo y el empoderamiento para abordar desafíos de frente.

Al disolver silos que a menudo causan fricción dentro de las organizaciones, los equipos autónomos y multifuncionales sirven como la piedra angular para unas operaciones DevOps fluidas. El resultado es un proceso simplificado que facilita respuestas rápidas a los cambios y fomenta una cultura de colaboración y responsabilidad. Esto no es solo una característica agradable de tener en la ingeniería moderna; es una estrategia fundamental para cualquier empresa que aspire a implementar DevOps con éxito.

Mejora continua

La mejora continua sirve como la base de DevOps, ofreciendo tanto ventajas técnicas como culturales que son indispensables para el desarrollo y operaciones de software moderno. Desde el punto de vista técnico, permite cualidades como la confiabilidad, adaptabilidad y eficiencia de los

procesos de entrega de software a través de un análisis continuo de métricas de rendimiento y la utilización de flujos de trabajo automatizados.

Estas características no solo hacen que el producto final sea robusto, sino que también contribuyen a la optimización de recursos y una entrega más rápida de características. Culturalmente, la mejora continua fomenta un entorno colaborativo, asegurando responsabilidad, alentando una cultura de aprendizaje y, en última instancia, ayudando a dismantelar los silos organizacionales. También se alinea estratégicamente con la mejora de la experiencia del desarrollador.

La importancia de la mejora continua se vuelve aún más clara cuando se ve a través del lente de los **ciclos de retroalimentación**, que actúan como el sistema nervioso del ciclo de vida de DevOps. Estos ciclos permiten el monitoreo en tiempo real y proporcionan métricas accionables que alimentan directamente las iniciativas de mejora continua. Al evaluar regularmente el rendimiento del sistema, el compromiso de los usuarios y otros **indicadores clave de rendimiento (KPI)**, las organizaciones pueden ajustar rápidamente sus estrategias y asegurar el éxito a largo plazo. Este enfoque dinámico y basado en datos es esencial en el panorama tecnológico acelerado de hoy, lo que convierte a la mejora continua no solo en una mejor práctica, sino en un requisito fundamental para la supervivencia competitiva.

Automatización

En los modelos tradicionales de desarrollo, el desarrollo y las operaciones han sido a menudo dos silos separados. Los desarrolladores se centraban en escribir código y construir aplicaciones, mientras que las operaciones se centraban en la implementación y el mantenimiento. Este enfoque desarticulado a menudo conducía a retrasos, ineficiencias y fricción entre los dos equipos.

La integración continua/entrega continua (CI/CD) emergió como el puente crucial para conectar estos dos mundos. Con CI, los cambios de código de múltiples colaboradores se fusionan frecuentemente en un repositorio compartido, donde se ejecutan pruebas automatizadas para detectar errores e inconsistencias lo más rápido posible. Esto fomenta un entorno más colaborativo al facilitar la identificación de problemas temprano en el ciclo de desarrollo. CD asegura que el código esté siempre listo para la implementación, eliminando largos períodos de congelación donde no se podían agregar nuevas características porque estaba pendiente un lanzamiento.

Los procesos manuales no solo son propensos a errores, sino que también actúan como un freno significativo en la velocidad y eficiencia, los mismos problemas que DevOps busca resolver. Antes de que la automatización fuera adoptada ampliamente, los administradores de sistemas configuraban servidores manualmente, un proceso tedioso y propenso a errores. Además, a los desarrolladores a menudo les resultaba difícil replicar el entorno de operaciones para pruebas, lo que conducía al síndrome notorio de **"funciona en mi máquina"**.

La automatización, en este contexto, no es un lujo sino una necesidad. Los scripts automatizados manejan tareas desde la prueba de código hasta la implementación, asegurando que el proceso sea lo más estandarizado posible, eliminando así muchos de los errores y retrasos asociados con la intervención manual.