

Integración continua: qué es y por qué deberías aprender a utilizarla

Martin Fowler es uno de los autores más reconocidos en el ámbito de prácticas ágiles:

“La integración continua es una práctica en la que miembros de un equipo integran su trabajo frecuentemente, típicamente cada persona integra al menos una vez al día, generando varias versiones por día. Cada versión ejecutable es verificada por un sistema automático de integración y pruebas para detectar errores de integración lo más rápido posible.”

Implementar esta práctica exitosamente involucra ciertos requisitos:

- ❑ Tener un repositorio maestro donde esté disponible todo el código fuente y del que cualquier integrante del equipo pueda obtenerlo.
- ❑ Automatizar el proceso de integración para que cualquier persona pueda generar una versión ejecutable del sistema a partir del código fuente.
- ❑ Automatizar las pruebas para que sea posible ejecutar la matriz (suite) de pruebas en cualquier momento con un solo comando.
- ❑ Asegurar que cualquiera puede obtener el ejecutable más reciente y tener la confianza de que es la mejor versión hasta el momento.

Todo esto requiere bastante disciplina y requiere un esfuerzo significativo para introducirlo a un proyecto, pero una vez habilitado es sencillo mantenerlo y brinda grandes beneficios.

Beneficios

Estos son algunos beneficios puntuales de la integración continua:

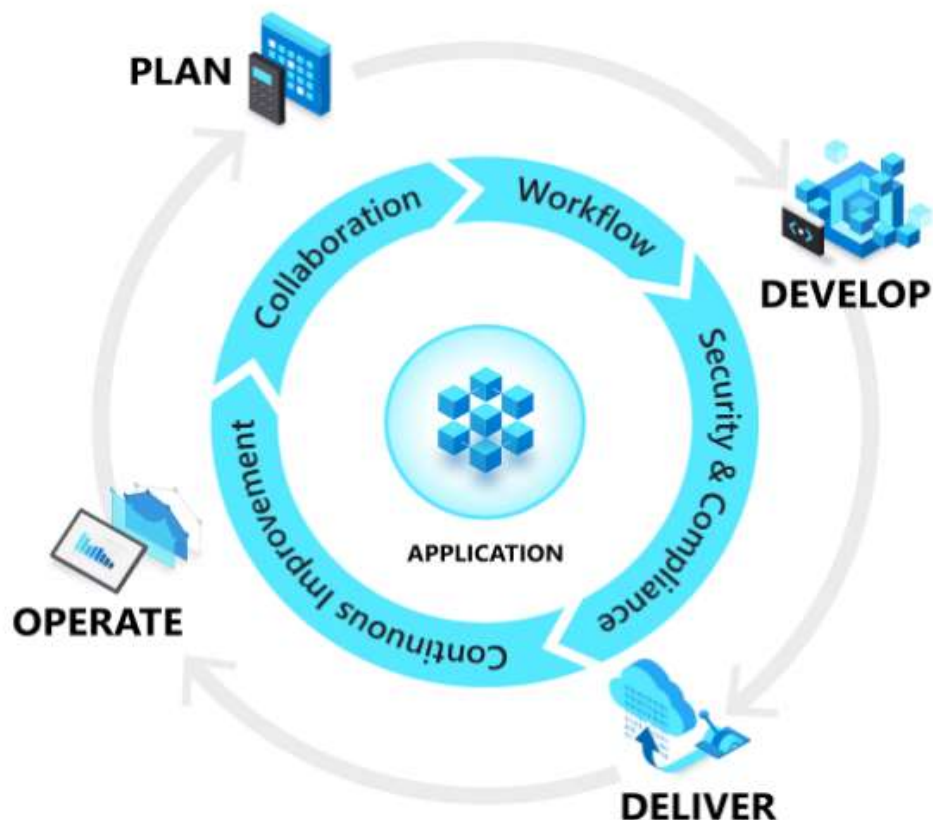
- ❑ Reducir problemas de integración.
- ❑ Mejorar la visibilidad del estatus del producto de software.
- ❑ Acelerar la detección de fallas.
- ❑ Disminuir el tiempo dedicado a depurar errores.
- ❑ Evitar la espera para averiguar si un código funciona.











¿Qué es DevOps?

DevOps une a las personas, los procesos y la tecnología para ofrecer a los clientes productos de más calidad en menos tiempo.

DevOps permite que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para crear productos mejores y más confiables.



Plan	Desarrollar	Entregar	Funcionamiento
<p>En la fase de planeamiento, los equipos de DevOps conciben, definen y describen las características y la funcionalidad de las aplicaciones y los sistemas que van a crear. Realizan un seguimiento del progreso tanto de forma general como de forma pormenorizada, desde tareas de un único producto hasta tareas que abarcan carteras de numerosos productos. La creación de registros de trabajo pendiente, el seguimiento de los errores, la administración del desarrollo de software ágil con Scrum, el uso de paneles Kanban y la visualización del progreso son algunas de las formas en las que los equipos de DevOps planean con agilidad y visibilidad.</p>	<p>La fase de desarrollo incluye todos los aspectos de la programación (escritura, pruebas, revisión e integración del código por parte de los miembros del equipo) y la compilación de ese código en artefactos de compilación que se pueden implementar en varios entornos. Los equipos de DevOps buscan innovar con rapidez sin sacrificar la calidad, la estabilidad ni la productividad. Para ello, utilizan herramientas muy productivas, automatizan los pasos cotidianos y manuales, e iteran el código en pequeños incrementos mediante pruebas automáticas e integración continua.</p>	<p>La entrega es el proceso de implementar aplicaciones en entornos de producción de un modo constante y confiable. La fase de entrega incluye también la implementación y la configuración de la infraestructura básica totalmente gobernada que constituye esos entornos.</p> <p>En la fase de entrega, los equipos definen un proceso de administración de versiones con fases de aprobación manual claras. También establecen puertas automáticas que mueven las aplicaciones de una fase a otra hasta que están disponibles para los clientes. La automatización de estos procesos hace que estén controlados y sean escalables y repetibles. De este modo, los equipos que practican DevOps pueden realizar entregas con facilidad, confianza y tranquilidad.</p>	<p>La fase de uso implica mantener y supervisar las aplicaciones, así como solucionar los posibles problemas, en los entornos de producción. Al adoptar prácticas de DevOps, los equipos trabajan para asegurar la confiabilidad, la alta disponibilidad y el objetivo de ningún tiempo de inactividad del sistema, al tiempo que refuerzan la seguridad y la gobernanza. Los equipos de DevOps buscan identificar los problemas antes de que afecten a la experiencia del cliente y mitigarlos rápidamente a medida que surgen. El mantenimiento de esta vigilancia requiere una telemetría muy completa, alertas que permitan tomar medidas y visibilidad total de las aplicaciones y del sistema subyacente.</p>

 Manage	 Plan	 Create	 Verify	 Package	 Secure	 Release	 Configure	 Monitor	 Protect
Subgroups	Issue Tracking	Source Code Management	Continuous Integration (CI)	Package Registry	SAST	Continuous Delivery	Auto DevOps	Runbooks	Container Scanning
Audit Events	Time Tracking	Code Review	Code Testing and Coverage	Container Registry	Secret Detection	Pages	Kubernetes Management	Metrics	Security Orchestration
Audit Reports	Boards	Wiki	Load Testing	Helm Chart Registry	Code Quality	Review Apps	Secrets Management	Incident Management	Container Host Security
Compliance Management	Epics	Static Site Editor	Browser Performance Testing	Dependency Proxy	DAST	Advanced Deployments	ChatOps	On-call Schedule Management	Container Network Security
Code Analytics	Roadmaps	Web IDE	Usability Testing	Release Evidence	Fuzz Testing	Feature Flags	Serverless	Logging	Product Analytics
DevOps Reports	Service Desk	Live Preview	Accessibility Testing	Git LFS	Dependency Scanning	Release Orchestration	Infrastructure as Code	Tracing	
Value Stream Management	Requirements Management	Snippets	Merge Trains		License Compliance		Cluster Cost Management	Error Tracking	
Insights	Quality Management				Vulnerability Management				
	Design Management								

“ Aunque la adopción de prácticas de DevOps automatiza y optimiza los procesos con tecnología, todo comienza con la cultura interna de la organización y con las personas que participan en ella. “



Colaboración, visibilidad y alineamiento

Una buena cultura de DevOps se distingue, entre otras cosas, por la colaboración entre los equipos, que comienza con la visibilidad. Diferentes equipos, como el de desarrollo o el de operaciones de TI, deben compartir entre sí sus procesos de DevOps, sus prioridades y sus preocupaciones. Estos equipos también deben planear juntos el trabajo y alinear sus objetivos y los indicadores del éxito en relación con el negocio.



Cambios en el ámbito y en la responsabilidad

A medida que los equipos se alinean, asumen y participan en más fases del ciclo de vida, no solo las que son principales para su rol. Por ejemplo, los desarrolladores asumen responsabilidad no solo por la innovación y la calidad establecidas en la fase de desarrollo, sino también por el rendimiento y la estabilidad que sus cambios producen en la fase de uso. Al mismo tiempo, los operadores de TI se aseguran de incluir la gobernanza, la seguridad y el cumplimiento normativo en las fases de planeamiento y desarrollo.



Ciclos de lanzamiento de versiones más cortos

Los equipos de DevOps mantienen la agilidad porque lanzan versiones de software en ciclos cortos. Los ciclos de lanzamiento de versiones más cortos facilitan el planeamiento y la administración de los riesgos, porque el progreso es incremental, lo que reduce el impacto en la estabilidad del sistema. El acortamiento de los ciclos de lanzamiento de versiones permite también a las organizaciones adaptarse y reaccionar a las necesidades cambiantes de los clientes y a la presión competitiva.



Aprendizaje continuo

Los equipos de DevOps de alto rendimiento establecen una mentalidad de crecimiento. Aceptan el fracaso y responden rápido a los errores, e incorporan lo que aprenden a los procesos, de modo que mejoran continuamente, aumentan la satisfacción del cliente y agilizan la innovación y la capacidad de adaptación al mercado de forma constante. DevOps es un recorrido, por lo que siempre hay espacio para crecer.

Prácticas de DevOps

Integración y entrega continuas (CI/CD)

La integración continua es una práctica de desarrollo de software en la que los desarrolladores fusionan mediante combinación los cambios de código en la rama de código principal con frecuencia. En la integración continua se utilizan pruebas automáticas, que se ejecutan cada vez que se hace "commit" de código nuevo. De este modo, el código de la rama principal siempre es estable.

La entrega continua es la implementación automática y frecuente de nuevas versiones de una aplicación en un entorno de producción. Al automatizar los pasos necesarios para la implementación, los equipos reducen los problemas que pueden surgir en ese proceso y permiten actualizaciones más frecuentes.

Cuando se utilizan estas dos prácticas, el proceso resultante es CI/CD, que incluye la automatización completa de todos los pasos, desde que se hace "commit" del código hasta que se implementa en el entorno de producción. La implementación de CI/CD permite a los equipos centrarse en la creación de código y elimina la carga y la posibilidad de errores humanos en los pasos cotidianos que se realizan manualmente. CI/CD también agiliza el proceso de implementación de nuevo código y reduce los riesgos que conlleva. Por tanto, las implementaciones son más frecuentes y se realizan en incrementos más pequeños, lo que ayuda a los equipos a ser más ágiles y más productivos, y a confiar más en el código que se ejecuta.

Control de versiones

Control de versiones es la práctica de administrar el código por versiones, haciendo un seguimiento de las revisiones y del historial de cambios para facilitar la revisión y la recuperación del código. Esta práctica suele implementarse con sistemas de control de versiones, como Git, que permite que varios desarrolladores colaboren para crear código. Estos sistemas proporcionan un proceso claro para fusionar mediante combinación los cambios en el código que tienen lugar en los mismos archivos, controlar los conflictos y revertir los cambios a estados anteriores.

El uso del control de versiones es una práctica de DevOps fundamental que ayuda a los equipos de desarrollo a trabajar juntos, dividir las tareas de programación entre los miembros del equipo y almacenar todo el código para poder recuperarlo fácilmente si fuese necesario.

El control de versiones es también un elemento necesario en otras prácticas, como la integración continua y la infraestructura como código.

Desarrollo ágil de software

El método ágil es un enfoque de desarrollo de software que hace hincapié en la colaboración en equipo, en los comentarios de los clientes y usuarios, y en una gran capacidad de adaptación a los cambios mediante ciclos cortos de lanzamiento de versiones. Los equipos que practican la metodología ágil proporcionan mejoras y cambios continuos a los clientes, recopilan sus comentarios y, después, aprenden y ajustan el software en función de lo que el cliente quiere y necesita. El método ágil es muy diferente a otros marcos más tradicionales, como el modelo en cascada, que incluye ciclos de lanzamiento de versiones largos definidos por fases secuenciales. Kanban y Scrum son dos marcos populares asociados al método ágil.

Infraestructura como código

La infraestructura como código define las topologías y los recursos del sistema de un modo descriptivo que permite a los equipos administrar esos recursos igual que lo harían con el código. Las diferentes versiones de esas definiciones se pueden almacenar en sistemas de control de versiones, donde se pueden revisar y revertir, de nuevo, igual que el código.

La práctica de la infraestructura como código permite a los equipos implementar recursos del sistema de un modo confiable, repetible y controlado. Además, la infraestructura como código ayuda a automatizar la implementación y reduce el riesgo de errores humanos, especialmente en entornos complejos de gran tamaño. Esta solución repetible y confiable para la implementación de entornos permite a los equipos mantener entornos de desarrollo y pruebas que sean idénticos al entorno de producción. De igual modo, la duplicación de entornos en otros centros de datos y en plataformas en la nube es más sencilla y más eficiente.

Administración de configuración

Administración de la configuración hace referencia a la administración del estado de los recursos de un sistema, incluidos los servidores, las máquinas virtuales y las bases de datos. El uso de herramientas de administración de la configuración permite a los equipos distribuir cambios de un modo controlado y sistemático, lo que reduce el riesgo de modificar la configuración del sistema. Los equipos utilizan herramientas de administración de la configuración para hacer un seguimiento del estado del sistema y evitar alteraciones en la configuración, que es cómo se desvía la configuración de un recurso del sistema a lo largo del tiempo del estado definido para él.

Al usarla junto con la infraestructura como código, resulta fácil elaborar plantillas y automatizar la definición y la configuración de sistemas, lo que permite a los equipos usar entornos complejos a gran escala.

Supervisión continua

Supervisión continua significa tener visibilidad total y en tiempo real del rendimiento y el estado de toda la pila de aplicaciones, desde la infraestructura subyacente donde se ejecutan las aplicaciones hasta los componentes de software de niveles superiores. La visibilidad consiste en la recopilación de datos de telemetría y metadatos, así como en el establecimiento de alertas para condiciones predefinidas que garanticen la atención de un operador. La telemetría incluye registros y datos de eventos recopilados de varias partes del sistema que se almacenan donde pueden analizarse y consultarse.

Los equipos de DevOps de alto rendimiento se aseguran de establecer alertas útiles que les permitan tomar medidas y recopilan datos de telemetría muy completos para obtener conclusiones a partir de enormes cantidades de datos. Estas conclusiones ayudan a los equipos a mitigar los problemas en tiempo real y a ver cómo mejorar la aplicación en futuros ciclos de desarrollo.

Visualización de la demostración x Home - Microsoft Azure x Overview dashboard - Overview x

https://dev.azure.com/sccsoft/SCC2.0/_dashboards/dashboard/5e8f1514-04b9-4636-93cb-80a6df6c5b58

Azure DevOps

sccsoft / SCC2.0 / Overview / Dashboards

SCC2.0 Team - Overview

Overview

Summary

Dashboards

Wiki

Boards

Repos

Pipelines

Test Plans

Artifacts

Welcome

Get started using Azure DevOps to make the most of your team dashboard.

Manage Work
Add work to your board

Collaborate on code
Add code to your repository

Continuously integrate
Automate your builds

Visualize progress
Learn how to add charts

Work assigned to Victor Portugal

Last time you checked, there were no results.

SCC2.0 Team

G S MQ JA

Work

Backlog

Board

Task board

Queries

Sprint Burndown

Set iteration dates to use this widget.

Set iteration dates

New Work Item

Enter title

Bug

Create

Open User Stories

0

Work items

Visual Studio

Open in Visual Studio
Requires Visual Studio 2013+

Get Visual Studio
See Visual Studio downloads

Project settings

https://dev.azure.com/sccsoft/SCC2.0/_boards/backlog/t/SCC2.0 Team?team=SCC2.0 Team

Type here to search

8:58 AM
5/22/2021

Jenkins cd

Blog Documentation Plugins Community Subprojects About English Download Search

Jenkins

Build great things at any scale

The leading open source automation server, Jenkins provides hundreds of plugins to support building, deploying and automating any project.

Documentation Download

Participate and Contribute!

Jenkins is a community-driven project. We invite everyone to join us and move it forward. Any contribution matters: code, documentation, localization, blog posts, artwork, meetups, and anything else. If you have five minutes or a few hours, you can help!

2021

JENKINS

What happened when DevOps met Covid? Find out in our 2021 survey [Read our results](#)

GitLab is the open DevOps platform

Iterate faster, innovate together: Our open DevOps platform is a single application for unparalleled collaboration, visibility, and development velocity.

Try GitLab for FREE

Watch a demo

