

# Despliegue continuo en 5 sencillos pasos

Por Eric Ries

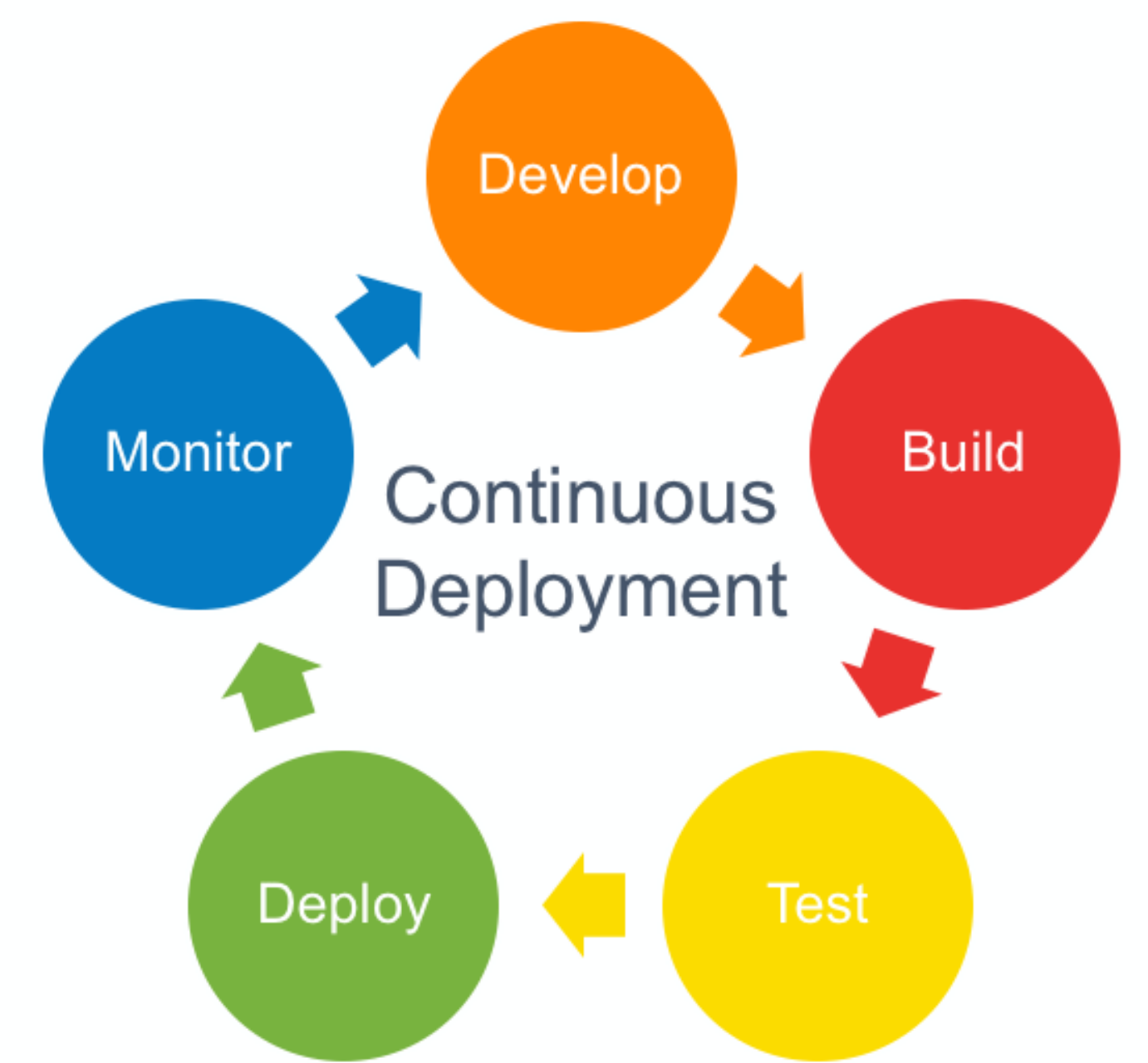
## Introducción

El despliegue continuo es un proceso mediante el cual todo el código que se escribe para una aplicación, se despliega inmediatamente en producción. El resultado es una reducción dramática del ciclo de tiempo y la liberación de la iniciativa individual.

El autor cree que el despliegue continuo, requiere de una sólida disciplina y puede mejorar en gran medida la calidad del software, al aplicar un conjunto riguroso de estándares a cada cambio, para evitar regresiones, interrupciones o daños a las métricas clave de negocio.

Al realizar unas simples inversiones y cambios en los procesos, cualquier equipo de desarrollo puede estar en camino a un despliegue continuo.

Estos son los 5 pasos que el autor propone para su correcta implementación:



## Desarrollo

1. Servidor de integración continua

2. Chequeo de commit con control de fuente

3. Script de implementación sencilla

4. Alerta en tiempo real

5. Análisis de causa raíz

Esta es la columna vertebral de la implementación continua. Se necesita de un lugar centralizado, donde todas las pruebas automatizadas (pruebas unitarias, pruebas funcionales, pruebas de integración, todo) se puedan ejecutar y monitorear en cada commit. Se agrega una nueva prueba automatizada cada vez que se solucione un error.

Se requiere un servidor de control de fuente, con un script de chequeo de commit. Cuando algo va mal con los sistemas, en cualquier lugar, a lo largo de la “línea de producción”, este script debe detener nuevos commit.

Lo más importante es que se tiene la oportunidad de ejecutar un código personalizado en el momento en que se envía un nuevo commit, pero antes de que el servidor lo acepte.

Esto configura el primer bucle de retroalimentación importante que se necesita para el despliegue continuo.

Es importante que se realice cada implementación de la misma manera, y se tenga un proceso claro y explícito sobre cómo hacerlo, para que pueda evolucionar con el tiempo.

Cuando se quiera subir el nuevo código a producción, requiere que todos usen este único mecanismo. Hay que mantenerlo manual, pero simple, para que todos sepan cómo usarlo. A medida que se obtiene experiencia se puede alcanzar la automatización en este paso.

No importa qué tan bueno sea el proceso de implementación, los errores aún pueden aparecer. La situación más molesta, es cuando los errores no se manifiestan hasta horas o días después de que el código ha sido desplegado. Para atrapar esos errores desagradables, se necesita una plataforma de monitoreo que permita saber cuándo las cosas han salido mal, y conseguir una persona involucrada en la depuración de ellos.

Los cinco “por qué” es una práctica potente, que es la fuerza que impulsará mejoras importantes en el proceso de desarrollo de manera gradual, un paso a la vez.

La idea es llegar siempre a la causa raíz de cualquier falla inesperada en el sistema. Este método recibe su nombre del proceso de preguntar “por qué” recursivamente, para descubrir la verdadera fuente de un problema dado. La forma en que cinco “por qué” funciona para habilitar despliegue continuo, es cuando agrega esta regla: cada vez que realice un análisis de causa raíz, realice una inversión proporcional en prevención, en cada uno de los cinco niveles que descubra, sin importar qué tan pequeño sea el problema.

## Conclusión

Siguiendo estos cinco pasos, no se obtendrá despliegue continuo en una noche. En las etapas iniciales, la mayoría del análisis de causa raíz, volverá al mismo problema: “aún no hemos invertido en prevenir eso”. Pero con paciencia y trabajo duro, cualquiera puede usar estas técnicas para eliminar inexorablemente el desperdicio de su proceso de desarrollo.

