Engineer

Entrega continua

mantener el software listo para publicarse con seguridad

Qué es

La entrega continua es la capacidad de publicar un cambio de software con seguridad cuando hace falta. Cada cambio sigue un recorrido que el equipo puede repetir y que deja pruebas de que puede pasar al siguiente paso. En producción puede haber un horario acordado, una persona responsable de aprobar la publicación o motivos operativos para esperar. Al tomar la decisión, el equipo ya dispone de un paquete probado y sabe cómo instalarlo en el destino.

En la integración continua, los desarrolladores incorporan cambios pequeños con frecuencia y comprueban que funcionan en el código compartido. La entrega continua incluye instalar el paquete resultante y demostrar que funciona en entornos adecuados. En el despliegue continuo, las modificaciones que cumplen las condiciones llegan de forma automática a producción. Una aprobación explícita de producción encaja con la entrega continua si se decide sobre la publicación de una versión que ya está preparada.

El despliegue y el lanzamiento también pueden ocurrir en momentos separados. El código puede estar funcionando en producción con una función desactivada. Más adelante, el equipo puede activarla para unas cuentas de prueba o para un grupo limitado de usuarios. Antes de cada paso conviene tener claro qué puede cambiar para el usuario y qué prueba permite comprobarlo. Una prueba de la instalación cubre una pregunta distinta de la prueba de la función recién activada.

Utilice el concepto en una decisión

Aplique «¿Qué es la entrega continua?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Cambios probados, publicados y reversibles, para que lleguen a sus clientes un martes por la tarde en vez de un viernes por la noche.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Despliegue GIS SaaS con GitHub Actions — despliegue, datos e informes automatizados

Despliegue GIS SaaS con GitHub Actions Póngase en contacto →

Por qué importa

Una publicación manual que depende de lo que recuerda una persona es difícil de traspasar. Un flujo compartido conserva el resultado de cada paso. La siguiente persona puede averiguar si falló una prueba, si el entorno no respondió o si queda pendiente una aprobación. Registra qué paquete está instalado mediante un identificador verificable. Si luego hay que investigar un problema, un nombre como `latest` aporta muy poca información sobre el código que en realidad estaba funcionando.

Un proceso pesado hace que el equipo acumule cambios hasta la siguiente oportunidad. Si el conjunto provoca un fallo, hay más causas que investigar. Cuando la ruta de entrega se puede repetir, es posible llevar cambios más pequeños a producción y observar el efecto de cada uno. La misma información ayuda a justificar una espera si otro servicio está en mantenimiento o si falta una comprobación relevante.

Una entrega puede incluir dependencias que se gestionan fuera del repositorio de la aplicación. Las migraciones de datos, las reglas de red y la configuración del inicio de sesión pueden tener equipos distintos a cargo. Identifica las combinaciones que deben funcionar juntas y pruébalas con ajustes representativos. Así queda un punto de partida para la persona que se haga cargo de la siguiente publicación o de la recuperación de un fallo.

Cómo se aplica en la práctica

El caso sobre los flujos de CI/CD con GitHub Actions recoge flujos separados para el control del código, las pruebas y el despliegue en cada entorno. Las imágenes de la interfaz, la API y la base de datos partieron de bases elegidas para cada carga. El caso muestra cómo un recorrido definido, con comprobaciones visibles, convirtió los lanzamientos en una tarea rutinaria. La selección de imágenes y los criterios de aprobación pueden variar en otros proyectos.

Supón que una aplicación añade una opción para descargar un informe. El flujo registra la revisión del código y genera un paquete identificado de forma inequívoca. Lo instala en un entorno de pruebas, donde una cuenta autorizada

descarga un informe con datos representativos. La prueba comprueba las columnas del archivo y quién puede acceder a ellas. La persona responsable consulta ese resultado junto al identificador del paquete. Si aprueba la entrega, el mismo paquete llega a producción con la configuración correspondiente. Una prueba breve confirma que funcionan tanto la descarga nueva como el endpoint que el usuario ya conocía.

Si la prueba de preproducción falla, el paquete queda disponible para investigar con una explicación del fallo. Si el despliegue llega a producción y falla allí, se puede consultar qué paquete se ejecutaba antes y qué indicador cambió. Ensaya todo el recorrido en un entorno controlado, incluso con una dependencia que se detiene adrede. Si hay cambios de base de datos, prueba también si la versión anterior puede leer el esquema en el estado intermedio.

Dónde surgen los problemas

Construir un paquete nuevo para cada entorno puede borrar el vínculo con las pruebas anteriores. Una dependencia puede haber cambiado o un compilador puede producir otros archivos a partir del mismo commit. Promueve el artefacto identificado que se probó siempre que los entornos lo puedan ejecutar. Cuando una plataforma necesita otra variante, guarda su identificador y exige las pruebas propias de ese destino.

Una aprobación puede convertirse en otro proceso de entrega. Quien decide necesita saber qué paquete se instalará, en qué entorno y qué resultados se han recogido. Si después debe repetir varios comandos en un orden secreto, hay una parte importante del despliegue que sigue dependiendo de una persona. Acordad qué preguntas requieren criterio humano y automatizad las operaciones repetidas que cualquier operador autorizado debería poder ejecutar.

Un resultado verde puede ser poco fiable si la prueba solo acierta tras varios intentos. Investiga las pruebas intermitentes y verifica que los indicadores de salud correspondan al comportamiento que importa. Si un cambio tiene un riesgo especial, deja un intervalo de observación antes de ampliar la exposición. Registra la condición que obliga a detenerse y quién puede decidirlo. Tras una interrupción, vuelve a un paquete identificado y a un estado comprobado en el entorno.

Lecturas adicionales

- DORA sobre entrega continua define la capacidad de publicar cuando hace falta.
- DORA sobre integración continua explica la incorporación frecuente de cambios y la información que devuelven las pruebas.
- DORA sobre automatización del despliegue describe las operaciones que pueden repetirse entre entornos.
- Controles de despliegue en GitHub Actions documenta entornos, trabajos y reglas de protección.

Glosario

Elija según sus restricciones reales

Describa la ruta de despliegue, los riesgos operativos y quiénes se encargan de las guardias. Podemos definir el alcance de un cambio con comprobaciones, una vía de recuperación y responsabilidades claras.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es la entrega continua?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/continuous-delivery/