

Integración Continua y Despliegue Continuo

por Alejandro Chacón

22/06/2023

Contents

1. Introducción	3
2. Integración Continua (CI)	3
2.1. Beneficios de la CI	3
3. Despliegue Continuo (CD)	4
3.1. Beneficios del CD.....	4
4. Conclusión.....	4
Referencias.....	5

1. Introducción

La industria del software actual es altamente competitiva y se mueve a velocidades vertiginosas, lo cual obliga a las compañías a buscar constantemente formas de expedir sus procesos de desarrollo, mejorar la calidad de su software y entregar dichos productos rápidamente al mercado. Para lograr este incremento de competencia que en primera instancia suena imposible, la industria ha recurrido a dos prácticas conocidas como la Integración Continua y el Despliegue Continuo (por sus siglas en inglés, CI y CD respectivamente). CI y CD son metodologías estrechamente relacionadas que se enfocan en simplificar el ciclo de vida del software, fomentando la colaboración, y garantizando un ciclo de liberación rápido y confiable. A continuación, se exploran los conceptos de las CI y CD, sus beneficios, dificultades y su impacto en las prácticas modernas del desarrollo de software.

2. Integración Continua (CI)

La idea general de la integración continua se puede dividir en dos partes. La primera, es simplemente que los miembros de un equipo de desarrollo integren sus cambios al código fuente de manera constante — diariamente, de forma que el software se esté integrando frecuentemente. La segunda parte, es la *verificación* automática del código fuente; por cada integración, el software debe ser compilado y probado por mecanismos automáticos para demostrar que los cambios pasan todas las pruebas y estándares definidos por el equipo de desarrollo. El propósito de esta automatización es detectar problemas de integración lo más pronto posible, facilitando que los desarrolladores produzcan software cohesivo rápidamente. Como se puede intuir, la CI es fundamental para cualquier proceso de desarrollo ágil moderno. (Fowler, n.d.)

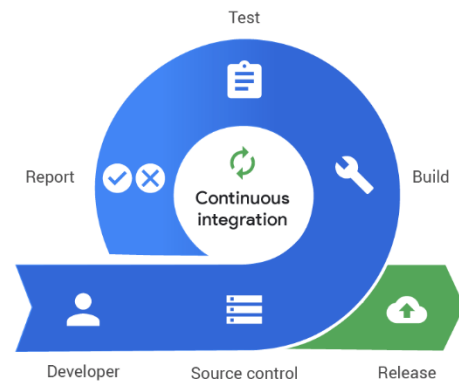


Ilustración 1. Integración Continua

2.1. Beneficios de la CI

Estudios hechos por el programa de Investigación y Evaluación DevOps (DORA, por sus siglas en inglés) han demostrado que el uso de CI mejora las siguientes métricas en el desarrollo de un sistema de software (Google Cloud, 2022):

1. Retroalimentación pronta y automatización de compilación y pruebas ayudan a agilizar la transición del código en una rama de desarrollo, a la rama de producción.
2. Frecuencia de despliegue: La automatización de la compilación y las pruebas, permiten la automatización del despliegue.
3. Restauración de servicios más ágil: pipelines automatizadas permiten que las correcciones de desplieguen a la rama de producción más rápidamente, reduciendo el tiempo promedio de resolución (MTTR).
4. Razón Cambio/Fallo: La automatización temprana de las pruebas incrementa enormemente el número de defectos que se inyectan a producción.

Por otro lado, también se deben mencionar los beneficios que el negocio obtiene gracias a la CI. (GitLab 2023)

1. Iteraciones rápidas: ya que las integraciones suceden múltiples veces al día, los cambios integrados son pequeños, haciéndolos más fáciles de administrar y liberar.
2. Encontrar problemas fácilmente: Los equipos pueden encontrar problemas en su código fuente ya que éste es administrado y probado en conjuntos pequeños.
3. Mejorar transparencia: La retroalimentación continua a través de las pruebas frecuentes ayuda a los desarrolladores ver en dónde se encuentran los errores.
4. Costos reducidos: La automatización reduce el tiempo invertido en tareas manuales.
5. Satisfacción del usuario: Menos errores en producción se traduce a una mejor experiencia del usuario.

3. Despliegue Continuo (CD)

El despliegue continuo es una estrategia de desarrollo de software en la que nuevas funcionalidades del software son liberadas al cliente frecuentemente mediante procesos de despliegue automatizados. Esta práctica, por ende, va de la mano con CI, pues sólo aquellas funciones que pasan por el pipeline automatizado de integración sin problema alguno pueden ser desplegados sin intervención humana. De esta forma, el tiempo entre la liberación del software y la retroalimentación del usuario se reduce enormemente; pues toda nueva funcionalidad puede llegar al mercado tan pronto termine de ser construida, en lugar de tener que esperar a que se libere en un día específico, o *release day* (Atlassian, n.d.)

3.1. Beneficios del CD

El despliegue continuo requiere que la organización tenga una política de pruebas robustas, pues la calidad de cada despliegue depende de ellas. Por otro lado, un despliegue continuo significa actualizaciones continuas a la documentación de un sistema, lo cual puede resultar engorroso en un principio, sin embargo, los beneficios ofrecidos por el CD lo valen:

1. Todo cambio al código fuente es liberado al usuario tan pronto se detecte en el repositorio, incrementando la velocidad de la retroalimentación del usuario.
2. El usuario ve un incremento de funcionalidades o correcciones de errores con mayor frecuencia, mejorando su experiencia con el sistema.
3. Ya que los cambios son constantes, también son más acotados. Esto hace que encontrar errores sea más fácil y rápido (similar a la CI), volviendo a cada liberación de software menos riesgosa.

Por su naturaleza, el CD va de la mano con la CI, es necesario que todos los cambios realizados al código fuente sean probados e integrados de forma constante y automática antes de ser liberados — la CI se encarga de esto.

4. Conclusión

La integración y la implementación continuas son metodologías poderosas que se han vuelto indispensables en el panorama moderno de desarrollo de software. Su capacidad para optimizar los procesos de desarrollo, mejorar la calidad del software y facilitar implementaciones rápidas y confiables las convierte en herramientas esenciales para las organizaciones que se esfuerzan por mantenerse

competitivas. Al adoptar CI y CD, las empresas pueden desbloquear el potencial para una innovación más rápida, una mayor satisfacción del cliente y mejores prácticas generales de entrega de software.

Referencias

Atlassian. (n.d.). *Continuous integration vs. delivery vs. deployment*.

<https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>

Fowler, M. (n.d.). *Continuous Integration*. martinowler.com.

<https://martinfowler.com/articles/continuousIntegration.html>

GitLab. (2023). What is continuous integration (CI)? *GitLab*. <https://about.gitlab.com/topics/ci-cd/benefits-continuous-integration/>

Google Cloud. (2022). *State of DevOps del 2022* | Google Cloud.

<https://cloud.google.com/devops/state-of-devops?hl=es>