



Universidad Veracruzana

CONTINUOUS INTEGRATION (CI) AND CONTINUOUS DELIVERY(CD)

ENSAYO DESARROLLO DE APLICACIONES

Domínguez Carmona José Javier

Mtro. Juan Luis López Herrera

22/Junio/2022

La integración continua (CI) y la entrega continua (CD) son procesos importantes en el desarrollo de software, ya que mediante ellos es posible automatizar y acelerar el proceso de desarrollo, generando que el software sea eficiente y rápido. Estas prácticas se han convertido en puntos clave la cultura de DevOps, cambiando la forma en que los equipos desarrollan y entregan software.

La integración continua es el proceso de integrar y monitorear continuamente los cambios realizados por múltiples desarrolladores en un entorno compartido. El objetivo principal de CI es identificar y resolver los problemas de integración de manera temprana y garantizar que el software se desarrolle y pruebe de manera consistente y uniforme. Esto se logra mediante la automatización, las pruebas unitarias y otros controles de calidad, cada vez que se realiza un cambio en el entorno compartido lo que optimiza el proceso que de forma manual conlleva gran inversión de tiempo (dependiendo del proyecto).

La CI se basa en el principio de respuesta rápida, lo que significa que los desarrolladores reciben comentarios inmediatos sobre la calidad de sus cambios y pueden resolver los problemas de manera oportuna. Esto ayuda a reducir la aparición de problemas posteriores al desarrollo, que pueden ser costosos y difíciles de solucionar.

Por otro lado, la entrega continua va más allá de la CI, automatizando y previniendo el proceso de liberación de software para producción. Con CD, se desea asegurar que los cambios aprobados se entreguen de manera rápida y confiable y estén listos para su publicación. Esto se logra mediante la automatización de los pasos de integración, como las pruebas de aceptación, las nuevas pruebas y la implementación en producción o pruebas previas a la producción.

La entrega continua se basa en el concepto de un enfoque definida y predecible que incluye todos los pasos necesarios para entregar software de calidad a los usuarios. Esto incluye el uso de técnicas como arquitectura de código, virtualización e integración para garantizar entornos de prueba y producción repetibles y consistentes.

La CI y el CD permite que los equipos de desarrollo colaboren rápidamente mientras mantienen la estabilidad y la calidad de las versiones de software. Mediante el uso de diferentes métodos de desarrollo y entrega, se puede reducir el error humano y acelerar el tiempo de comercialización.

Sin embargo, implementar CI/CD es más que una simple herramienta. Requiere una visión única, claridad y mejora continua de todo el equipo de desarrollo. Además, es importante contar con buenos procedimientos de control de versiones, pruebas consistentes y un buen diseño de software, aunado al control bien definido de los pasos estructurados de integración y despliegue.

Esta metodología de trabajo es esencial para el desarrollo de software moderno. Al automatizar la producción, las pruebas y la entrega, mejoramos el cronograma, la calidad y la velocidad. Aunque esta implementación requiere un cambio en la cultura y las habilidades

del equipo de desarrollo, pero los beneficios son significativos ya que permite una entrega rápida, confiable y continua de sistemas de alta calidad a los usuarios finales.

Además de los puntos mencionados anteriormente, para implementar CI/CD correctamente se deben de usar las herramientas adecuadas. Hay muchas herramientas que pueden manejar diferentes aspectos, como sistemas de control de versiones (v. g. Git), sistemas de compilación (como Jenkins, Travis CI), instrumentos de administración (como Maven o npm), mecanismos de pruebas (como JUnit o Selenium), y frameworks de implementación e integración como Docker o Kubernetes.

La seguridad también es una consideración importante por lo que es necesario tener siempre en cuenta las mejores prácticas, un ejemplo de ello es el análisis de riesgo de las dependencias de software como lo son los conocidos Components of the shelf (COTS), la autenticación y autorización adecuadas en entornos de implementación, y la protección de claves y credenciales de acceso.

Cada vez que los equipos de desarrollo recopilan información y reciben comentarios de los usuarios finales, es necesario perfeccionar y mejorar continuamente las entregas y los procesos relacionados. Esto implica la supervisión y el análisis de métricas clave como los plazos de entrega, los tiempos de respuesta de las pruebas y la estabilidad del sistema para identificar áreas de mejora y optimización.

Finalmente, la integración y el despliegue continuos son técnicas importantes en el desarrollo de software de hoy en día. Al automatizar compilaciones, pruebas y versiones, podemos lograr un desarrollo de software más eficiente y rápido. Sin embargo, su uso requiere no solo las herramientas adecuadas, sino también la colaboración del equipo de desarrollo, la claridad y una cultura de cambio constante. Al adoptar y optimizar este enfoque, los equipos pueden entregar software de alta calidad de manera rápida, confiable y consistente, lo que les permite responder rápidamente a los cambios comerciales y las necesidades de los clientes.

Bibliografía

Humble, J., & Farley, D. (2010). Continuous delivery: reliable software releases through build, test, and deployment automation. Pearson Education.

Mercedes Brito, J. C. (2022). Integración y entrega continua (CI/CD) con Jenkins.