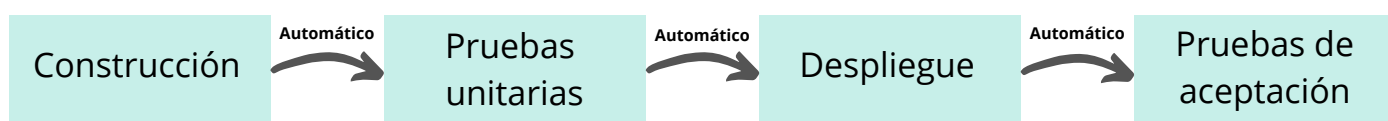


INTEGRACIÓN CONTINUA

Introducción

El primer paso para entregar un software consistente y de alta calidad es la Integración Continua (CI). Esta trata de garantizar que el software se encuentre en un estado de implementación en todo momento, asegurando una buena calidad del código.



Ventajas

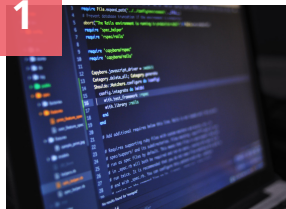
- Permite mitigar riesgos. En todo momento se sabe qué es lo que funciona o no en el sistema.
- A pesar de que la CI no evita los bugs, los mismos son más simples de encontrar y eliminar. Existen menos errores tanto en producción como en proceso.
- Es una forma de transparencia extrema. Asegura que todos los interesados (stakeholders) pueden monitorear, comprometerse y contribuir positivamente para la evolución del proyecto de software sin interrumpir al equipo con reuniones constantes.

Desventajas

- Las pruebas unitarias no deben tener errores, de lo contrario, pueden provocar grandes daños al ejecutarse.
- El cambio en la metodología de trabajo de todos los miembros puede requerir mucho tiempo.
- Inicialmente el costo de inversión puede resultar elevado.
- La automatización requiere tiempo para programar las pruebas automáticas para cada nueva característica del software.

Actividades de la CI

1



Construir el sistema

Incluye compilar el código y potencialmente reconstruir pruebas unitarias.

2



Ejecutar pruebas unitarias

Las pruebas que se ejecuten van a estar determinadas por el alcance de la compilación y el tamaño del sistema.

3



Realizar análisis estático

Comprueba si hay defectos en el código, buscando problemas de seguridad o de estilo de codificación.

Características

Utiliza TDD (Test-Driven Development) como técnica de desarrollo.

Fusiona continuamente las actualizaciones del código fuente de todos los desarrolladores de un equipo en una línea principal compartida.

Es una técnica automatizada.

Se traduce en código de mejor calidad.

El servidor de CI ayuda a reforzar el hábito de testear minuciosamente el código antes de enviarlo.

Permite acceso al estado en tiempo real del software.

Posibilita la realización de tests unitarios.

Puede utilizar un servidor de CI pero la técnica no se define por su utilización.

Notifica a los interesados (stakeholders) de los cambios, tanto éxitos como fracasos.

Permite obtener un build automatizado y funcionando del software todo el tiempo.

Conclusión

La Integración Continua como técnica automatizada brinda múltiples beneficios, como la disminución de riesgos en los proyectos de software, logrando mantener un nivel de transparencia y calidad ideal en los mismos.

Referencias

Rossel Sander - Continuous Integration, Delivery and Deployment

Martin Fowler - Continuous Integration Certification

Scott Wambler - Agile Testing and Quality Strategies: Discipline Over Rhetoric

C. Aaron Cois - Continuous Integration in DevOps

