

INTEGRACIÓN CONTINUA



1 INTRODUCCIÓN

Es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas.

2 AUTOMATIZACIÓN DE TESTS

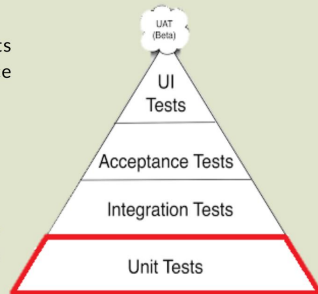
Es necesario automatizar los diferentes tests que se aplicarán en cada cambio que se realice en la mainline.

Tests unitarios: Verifican el comportamiento de métodos o funciones individuales.

Tests de integración: Se asegura que múltiples componentes se comporten de manera correcta en conjunto.

Tests de aceptación: Son similares a los test de integración pero ponen su foco en el caso de negocio.

Tests de interfaz de usuario(UI): Aseguran que la aplicación funcione correctamente desde la perspectiva del usuario.



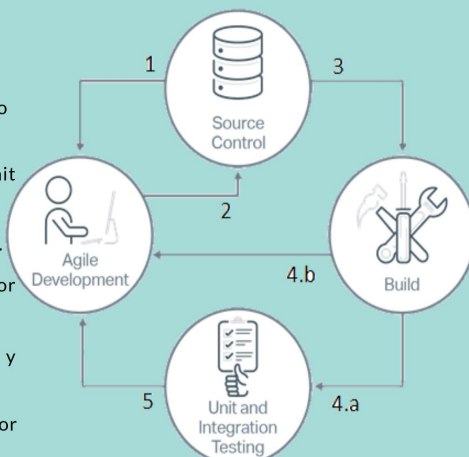
4 BENEFICIOS

- Mejora la calidad del código.
- Detección de errores más rápido y más fácil.
- Reduce tareas repetitivas y manuales.
- Puede crear versiones de prueba en cualquier momento.
- Completa visibilidad del proyecto.
- Mayor confianza y seguridad en el equipo de trabajo.



3 PASOS

- 1) El desarrollador toma el código del repositorio.
- 2) El desarrollador realiza commit a cambios.
- 3) El servidor de CI inicia la build.
- 4.a) Notificación a desarrollador sobre fallos en build.
- 4.b) Se realizan tests unitarios y de integración de la build.
- 5) Notificación a desarrollador sobre cambios en la build.



5 DESAFÍOS PARA SU ADOPCIÓN

- **Falta de una estrategia de prueba adecuada:** surge al no poder automatizar todo tipo de pruebas.
- **Mala calidad de la prueba:** Impidiendo el proceso de implementación y reduciendo la confianza.
- **Fusión de conflictos:** Integración de código que causa cuello de botella. Se atribuyen principalmente a un diseño altamente acoplado.
- **Resistencia general al cambio:** puede requerir la adopción de una nueva forma de trabajo para algunos miembros del equipo.
- **Costo:** puede incluir actualización en los recursos, capacitación y entrenamiento de los integrantes del equipo.



6 CONCLUSIÓN



Si logramos una buena implementación de la integración continua vamos a reducir sustancialmente los errores y la velocidad de resolución.

La utilización de repositorios nos permite mantener la información acomodada y controlar el versionado, evitando problemas a la hora de querer realizar cambios.

Con el conocimiento adquirido, podremos tener una base para mejorar la calidad de futuros productos en los que trabajaremos, incorporando los beneficios de esta práctica.



7 METODOLOGÍA

Buscamos y analizamos información sobre la Integración Continua. Tratamos de que los contenidos elegidos no coincidan en su totalidad con los otros grupos que tienen este tema. Seleccionamos la información.

Buscamos y analizamos información sobre cómo se confecciona un póster científico y sus partes. Con la información seleccionada procedimos a realizarlo.



UNIVERSIDAD TECNOLÓGICA NACIONAL
Facultad Regional Córdoba

Ingeniería en Sistemas de Información
Ingeniería de Software

DOCENTES

Meles, Silvia Judith
Titular

Massano, María Cecilia
JTP

Robles, Joaquín Leonel
Ayudante

AUTORES

Bordón, Eliana
elianabordon@gmail.com

Cruz, Karen
karenyanina.cruz@gmail.com

Konicoff, Sebastián
sebakonicoff@gmail.com

López Ríos, María Sol
lopezrios.sol@gmail.com

Palacio, Priscila
prii.palacio@gmail.com

Salaberri, Marcelo
marcelo4431@gmail.com



SCAN ME