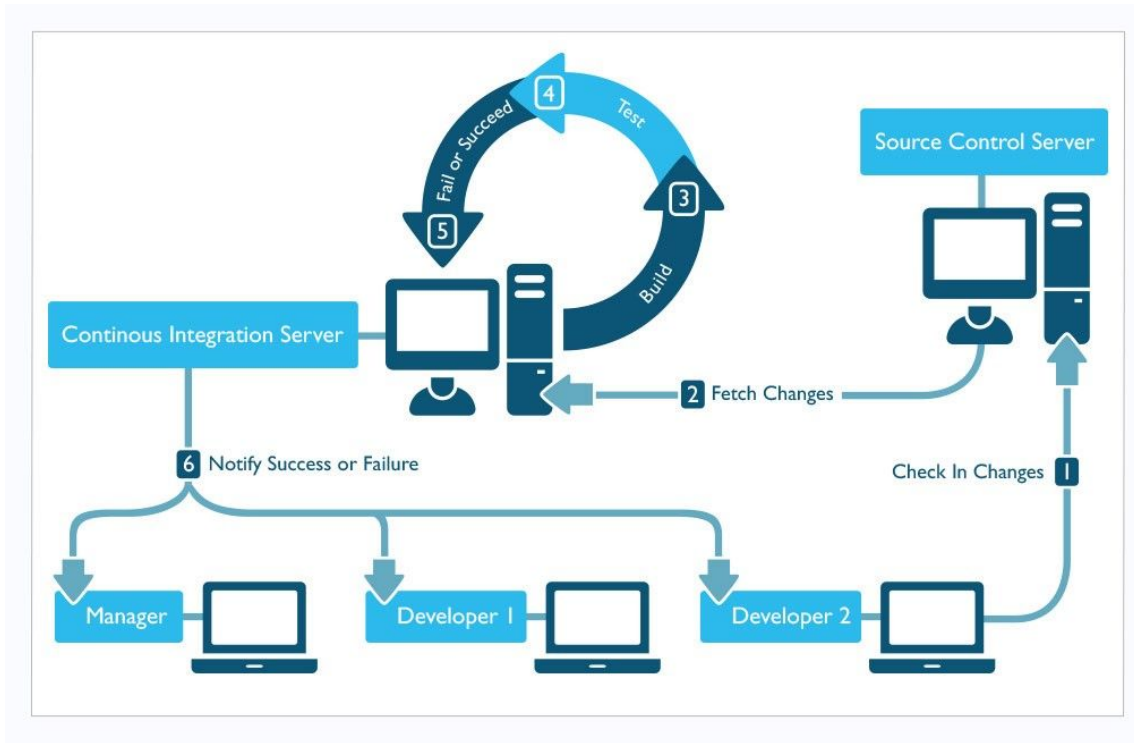


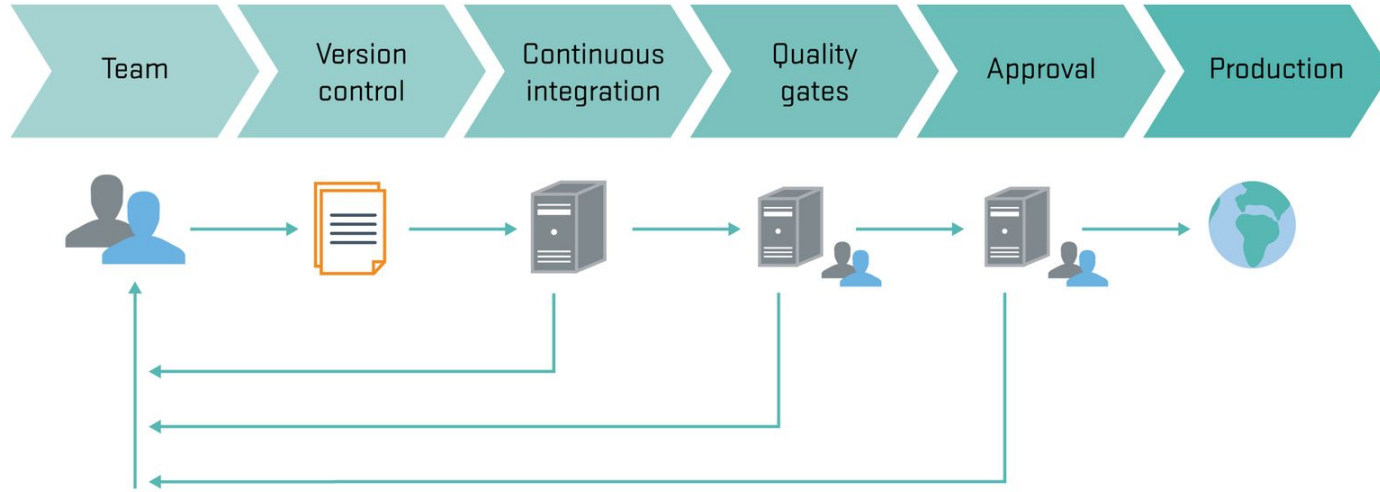
Continuous Integration
Continuous Delivery
Continuous Deployment

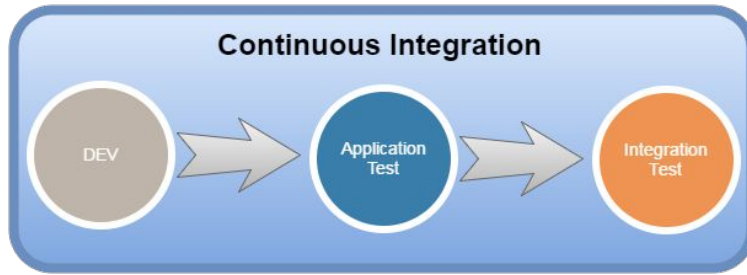


Build and Test Automation

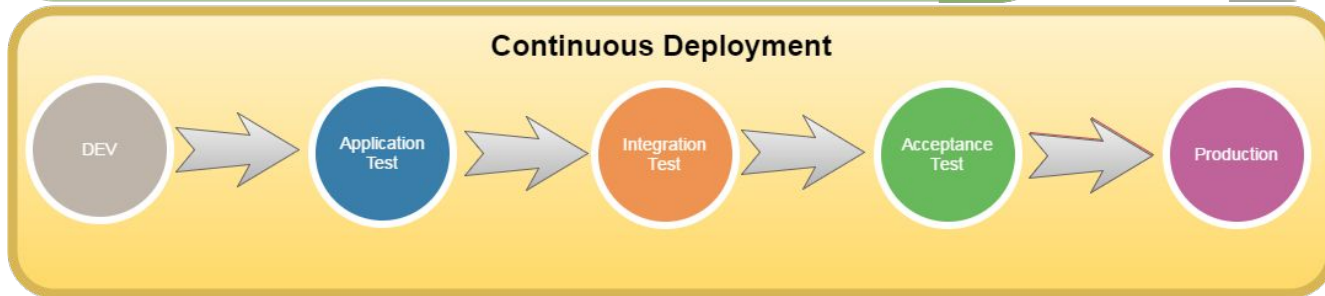
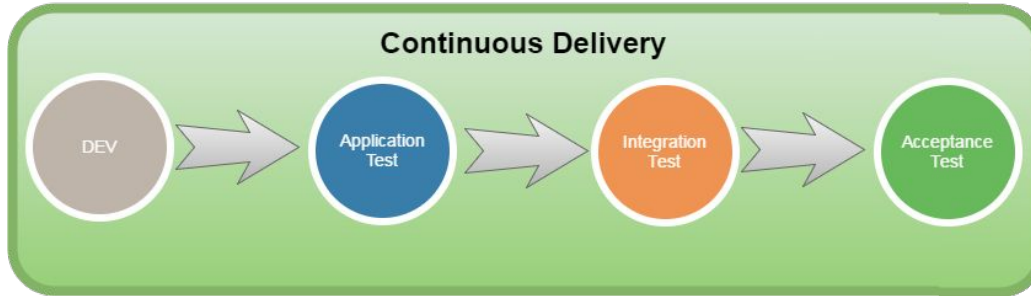


CI / CD

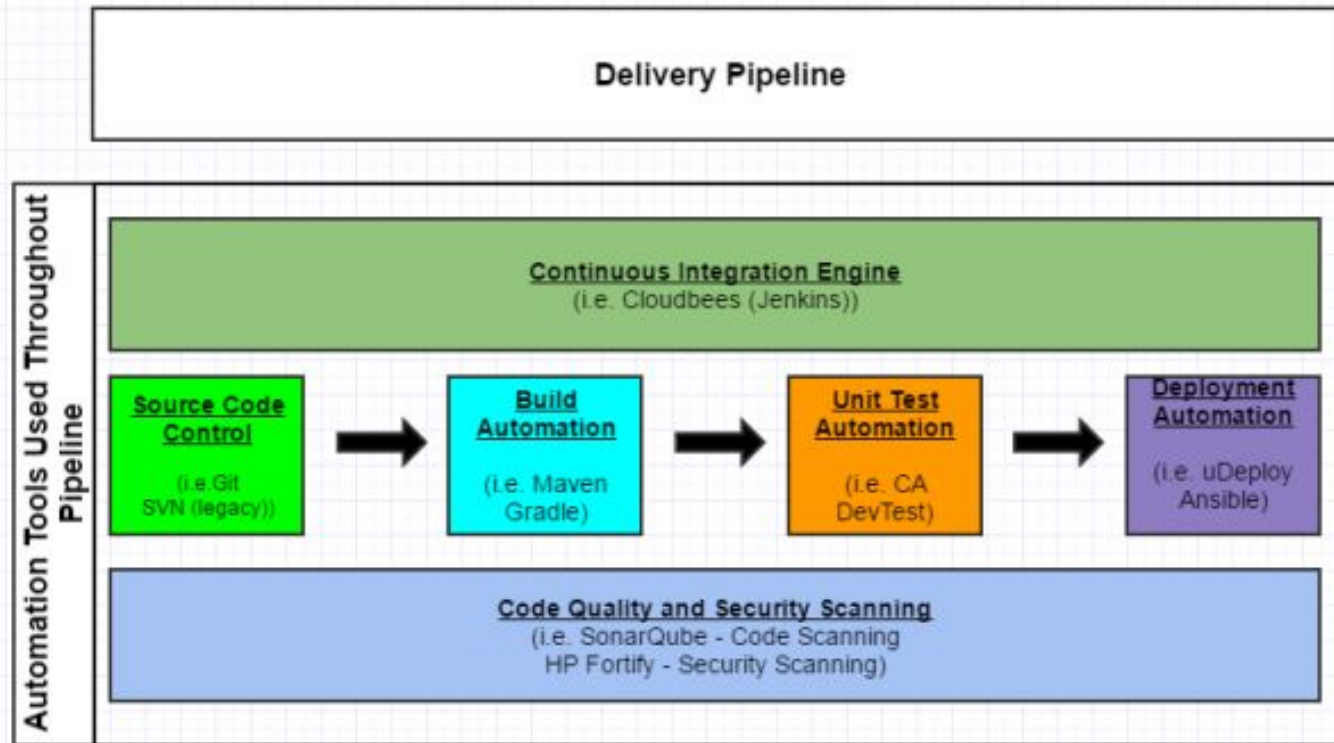





Automatic trigger



Continuous Integration



Fuente: <https://dzone.com>

1. Administración del código fuente

El código es administrado en un sistema de control de versiones con las siguientes ventajas:

- Seguimiento,
- Mantenimiento,
- Versionamiento,
- Auditoría.

Herramientas: GIT - MERCURIAL - SVN - CVS

2. Construcción Automática

El equipo debe poder compilar y construir el código de manera automática.

Ventajas:

- Reproducible
- Limpia
- Verificable

Permite verificar que el código es construible antes de enviarlo a pruebas o a producción.

Herramientas: Maven - Gradle

3. Automatización de pruebas unitarias

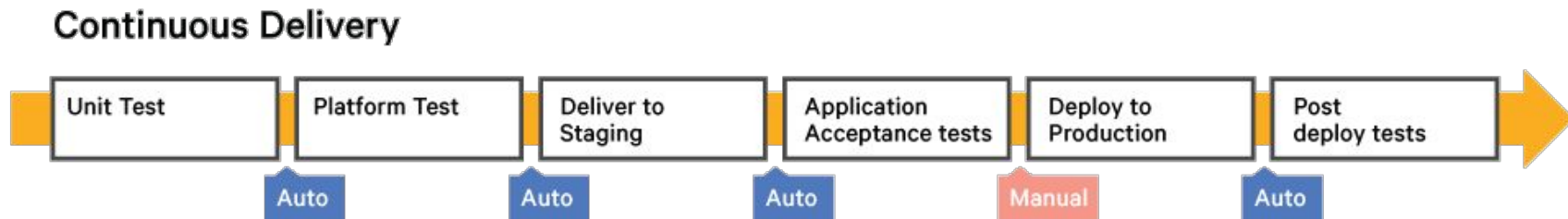
Verifica que la construcción de bloques pequeños de código funciona bien.

Herramientas: junit, ca devtests, quicktheories.



Continuous Delivery

Se despliega el código en uno o varios ambientes de prueba para la realización de pruebas de calidad del código y pruebas de funcionalidad



1. Análisis estático y Malos olores

Conjunto de verificaciones en el código que permiten revisar algunas propiedades del código que pueden presentar problemas en el futuro y se pueden corregir principalmente refactorizando:

- Código duplicado
- Método grande
- Clase grande
- Muchos parámetros
- Monopolio de otra clase
- Intimidación de otra clase
- Herencia rechazada (Liskov)
- Clase perezosa o inútil
- Complejidad artificial
- Identificadores muy largos
- Obsesión por primitivos (literales)
- Super callbacks

Herramientas: Codacy, SoneQube, ESLint

2. Ambiente de Pruebas

En algunos procesos de desarrollo se despliega en ambientes de pruebas donde se realizan pruebas funcionales

Herramientas: selenium, katalon studio, uft, etc.

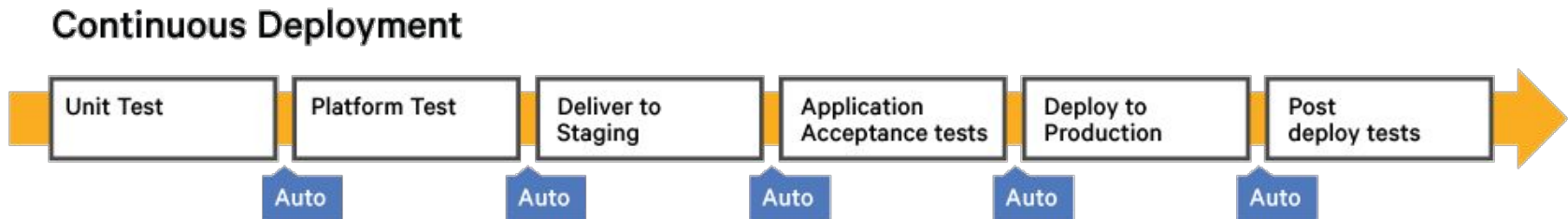
3. Aceptación

Se determina si el software cumple las condiciones bajo las cuales puede ser desplegado en producción



Continuous Deployment

Si el software pasa las condiciones de aceptación, el código es desplegado en producción; adicionalmente se realizan algunas pruebas y condiciones para determinar que el código se desplegó correctamente.



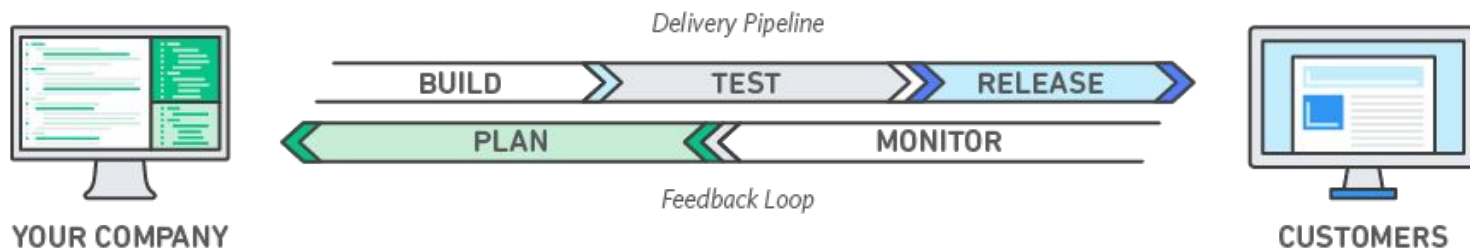
Beneficios

- Reducción de riesgos en el proceso y al momento del despliegue
- Proceso creíble y confiable
- Retroalimentación del usuario de manera oportuna



DevOps

Combinación de filosofías, prácticas y herramientas culturales que aumentan la capacidad de una organización para entregar aplicaciones y servicios a gran velocidad: evolucionando y mejorando productos a un ritmo más rápido que las organizaciones que utilizan procesos tradicionales de desarrollo de software y gestión de infraestructura. Esta velocidad permite a las organizaciones servir mejor a sus clientes y competir de manera más efectiva en el mercado. Amazon Web Services (AWS)



DevOps

- Integración de los equipos de desarrollo y operaciones
- Los ingenieros trabajan en todo el ciclo de vida de la aplicación: desde desarrollo, y pruebas hasta implementación y operaciones.
- Rango de habilidades no limitadas a una sola función.
- Integración de los equipos de seguridad y aseguramiento de la calidad posiblemente durante todo el ciclo de vida de la aplicación.

DevOps

Los equipos integrados usan:

- prácticas para automatizar procesos que históricamente han sido manuales y lentos.
- tecnología y herramientas que les ayudan a operar y desarrollar aplicaciones de forma rápida y confiable.
- herramientas que ayudan a los ingenieros a realizar de manera independiente algunas tareas que normalmente requerirían ayuda de otros equipos

IT As A Service

- Los servicios escalan automáticamente a medida que se incrementa la demanda
- El cobro se hace por los servicios utilizados en lugar de un cobro fijo mensual
- Flexibilidad y escalabilidad utilizando los servicios en la nube y no en servidores locales



Diferentes modelos de AAS: IAAS-PAAS-SAAS

