

CONTEXTO

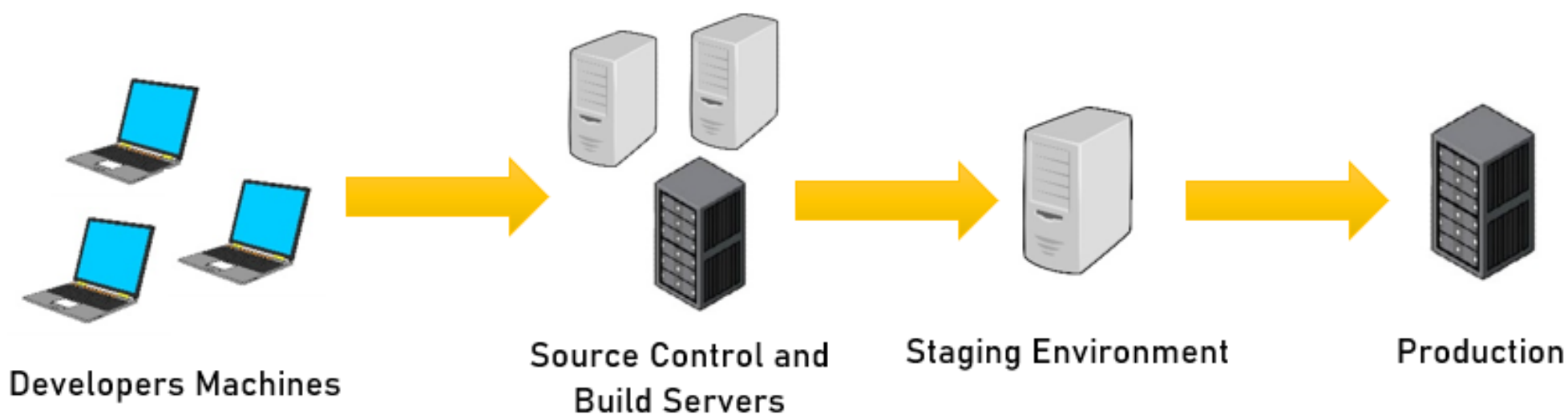
Este poster es realizado por los alumnos Bassano Joaquín, Camargo Ignacio, Cisneros Fernando, Giorda Matías, Sanchez Matías y Reyes Mateo, pertenecientes al grupo 3 del curso 4k1, de la cátedra de Ingeniería de Software, en la Universidad Tecnológica Nacional Facultad Regional Córdoba, año 2020.

INTRODUCCIÓN

Continuous Delivery es una disciplina aplicada en el desarrollo de software en donde se realizan, de forma automática, cambios de todo tipo en el código, incluidas nuevas funciones, cambios de configuración, corrección de errores y pruebas. El foco central de esta práctica, resulta en la automatización de las pruebas de aceptación de usuario, de modo de lograr ya una versión de software que se pasará a producción en forma manual.

MATERIALES Y MÉTODOS

Antes de comenzar a emplear Continuous delivery, es necesario elegir el conjunto de herramientas que se utilizarán y una configuración de requisitos previos para las mismas. Considerando un escenario base, describimos los materiales/componentes que se necesitan para la estructura:

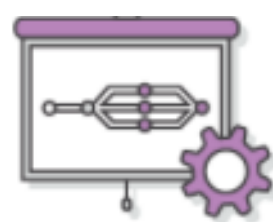


Luego de tener una estructura, podemos elegir entre dos estrategias básicas de entrega:

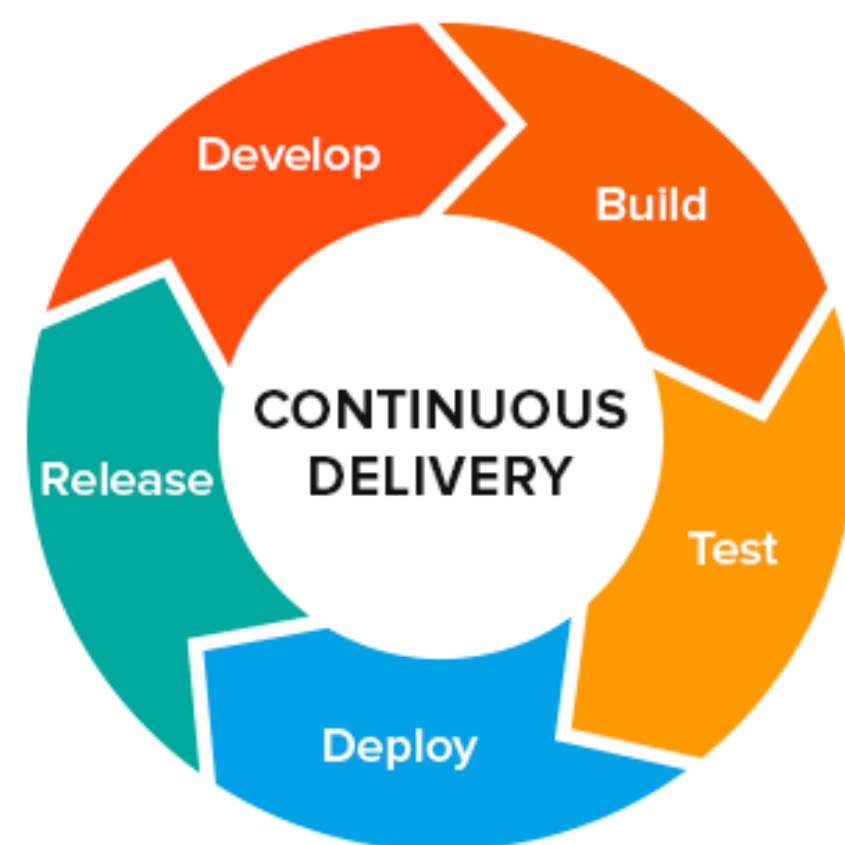
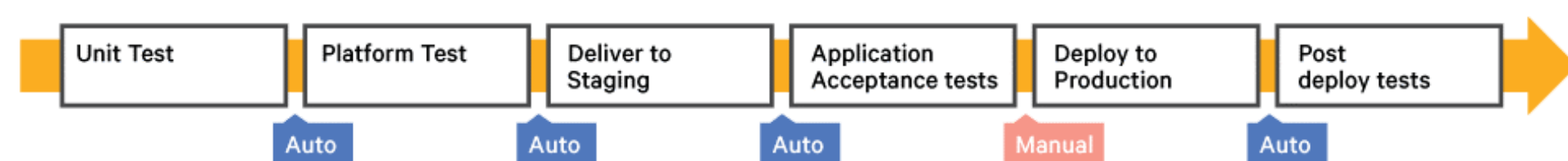
- Entregar en el check-in: Proporciona la gratificación instantánea. Las nuevas características se pueden compilar y probar en tan solo 1 o 2 minutos. Trae consigo varias desventajas.
- Entregar en el temporizador: Ciclos de compilación y entregas con momentos diarios predefinidos. Puede generar estrés a los desarrolladores.

RESULTADOS

- Automatiza el proceso de publicación de software
- Mejora la productividad de desarrollo
- Encuentra y arregla los errores con mayor rapidez
- Entrega las actualizaciones con mayor rapidez



Continuous Delivery



CONCLUSIÓN

Continuous Delivery es una disciplina que forma parte de la evolución de la Gestión de Configuración del Software, la cual da una especial importancia al Testing, haciendo que las implementaciones y cambios en el código pasen por las pruebas de aceptación del usuario de forma automatizada.

Como beneficio, esto permite que dichas pruebas se realicen en un entorno que no será el ambiente donde el software va a funcionar. A partir de ello, se desplegarán luego las implementaciones en el entorno de producción en forma manual.

REFERENCIAS Y AGRADECIMIENTOS

La bibliografía utilizada es la siguiente:

<https://www.infoq.com/articles/Continuous-Delivery-Patterns/>
<https://martinfowler.com/bliki/ContinuousDelivery.html>

<https://continuousdelivery.com/>
<https://aws.amazon.com/es/devops/continuous-delivery/>

Agradecemos a la Ingeniera Judith Meles por su guía y consejos durante las clases, y a la página web canva que nos permitió diseñar el poster de manera intuitiva y creativa.