

CONTINUOUS DELIVERY

Samuel Rodríguez Ares (U0271612)

Alba Guerrero García (U0266007)

Victoria Álvarez Sordo (U0266618)

TABLA DE CONTENIDOS

01

INTRODUCCIÓN

Qué es Continuous Delivery
y cómo se originó

02

COMPARATIVA

con otras prácticas
relacionadas

03

PRINCIPIOS

Objetivos principales
de Continuous Delivery

04

CIMENTOS

En qué se basa
Continuous Delivery

05

BENEFICIOS

Ventajas que supone
para la arquitectura

06

HERRAMIENTAS

para su implementación

07

ERRORES COMUNES

al aplicar
Continuous Delivery

08

CONCLUSIÓN

Síntesis de ideas
expuestas

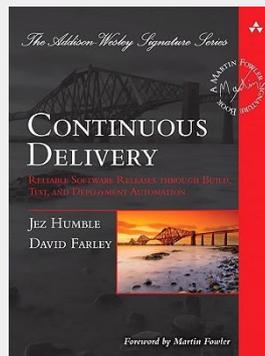
09

REFERENCIAS

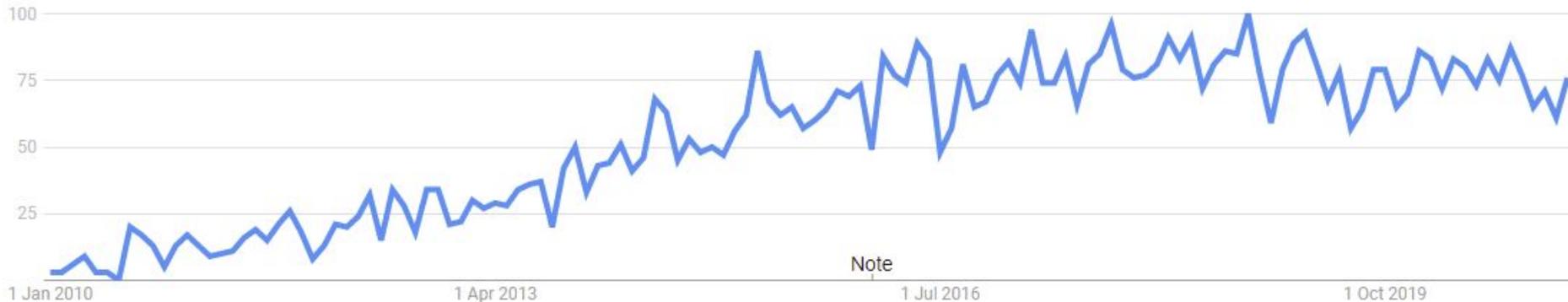
Videos explicativos y
webgrafía consultada

INTRODUCCIÓN > HISTORIA

- Tiene sus inicios alrededor del año **2010**
- Se vuelve más importante tras la publicación del libro *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*
- Sus beneficios animan a ser **cada vez más utilizado** día a día por empresas



Fuente: Google Trends



INTRODUCCIÓN > CONCEPTO



EXTIENDE LA INTEGRACIÓN CONTINUA

Mayor número de pasos automatizados



DISPONIBILIDAD PARA PRODUCCIÓN

Siempre existe un artefacto listo para entregar y mostrar al cliente



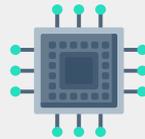
PRUEBAS AUTOMATIZADAS

Ejecutadas al producir cambios



PROGRAMAR LANZAMIENTOS

Diarios (Nightly), Semanales (Weekly)...



MENOR INTERACCIÓN HUMANA

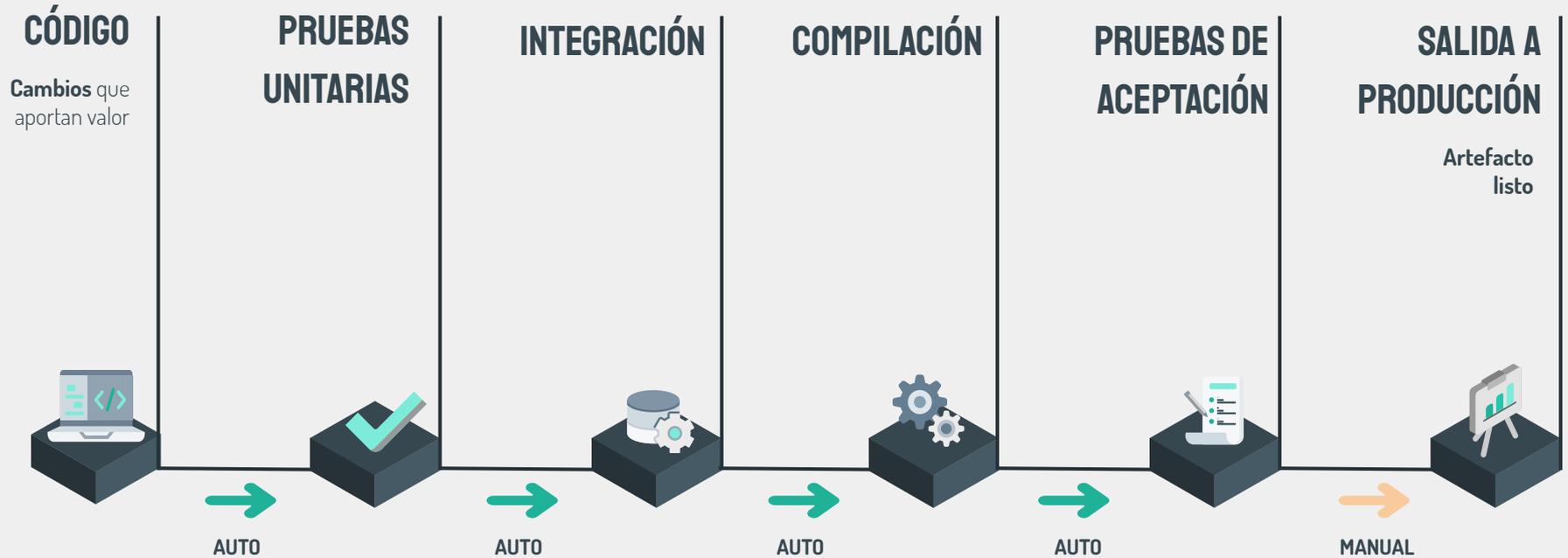
Los desarrolladores toman la única decisión de lanzar a producción



MEJOR ENTORNO DE FEEDBACK

Retroalimentación más continua por parte del cliente y otros stakeholders

INTRODUCCIÓN > CONCEPTO



COMPARATIVA

INTEGRACIÓN CONTINUA



ENTREGA CONTINUA



DESPLIEGUE CONTINUO



PRINCIPIOS



CALIDAD

Pruebas automatizadas para detectar y corregir errores.



PERÍODOS BREVES

Trabajar en ciclos cortos proporciona feedback temprano, mayor eficiencia



DISTRIBUCIÓN DEL TRABAJO

Delegar en las máquina el trabajo más repetitivo y de menor valor



MEJORA CONTINUA

Perspectiva de trabajo no conformista, intención de mejora constante



RESPONSABILIDAD CONJUNTA

Todo el equipo es responsable del software a construir

CIMIENTOS

DEPLOYMENT PIPELINE

Patrón clave

GESTIÓN DE CONFIGURACIÓN

TRAZABILIDAD

Determinar dependencias y componentes

REPRODUCIBILIDAD

Creación de nuevos entornos a partir de la misma base

INTEGRACIÓN CONTINUA

Someter el código a diferentes pruebas tras realizar cambios "valiosos"

PRUEBA CONTINUA

Ejecución de múltiples pruebas a lo largo del proceso de entrega

UNITARIAS

ACEPTACIÓN

USABILIDAD

BENEFICIOS PARA LA ARQUITECTURA



AUTOMATIZACIÓN

Mayor número de pasos automatizados



LIBERACIÓN DE RESPONSABILIDADES

Menor carga de trabajo a largo plazo



MEJORAS EN DEPURACIÓN

Simplificación a la hora de encontrar errores



RAPIDEZ DE ENTREGA

Actualizaciones preparadas y solicitables por el cliente en cualquier momento



HERRAMIENTAS DE IMPLEMENTACIÓN

ERRORES COMUNES



COMPILACIONES ROTAS

No arreglar errores antes de agregar código



ESTADO FINAL

Tratar la entrega continua como un estado final, un objetivo en sí mismo



PRUEBAS SIMPLES

No escribir pruebas suficientemente exhaustivas



ENTREGAS NO FIABLES

Realizar entregas sin trazabilidad de requisitos comprobada e infrecuentemente



PENSAR EXCESIVAMENTE

Dedicar mucho tiempo decidiendo los productos a utilizar



PRUEBAS FALLIDAS

Cerrarlas sin determinar su causa y localización

CONCLUSIONES

- Práctica de software en crecimiento desde 2010
- Implementarla puede resultar un desafío en un principio
- Es normal cometer errores
- Mejora la eficiencia del equipo a largo plazo
- Mejor entorno de retroalimentación



REFERENCIAS

VIDEOS EXPLICATIVOS

- [Qué es la Entrega Continua](#)
- [Entrega Continua vs Despliegue Continuo](#)
- [Qué es la Integración Continua](#)

WEBGRAFÍA

- <https://www.softwaretestingmaterial.com/must-know-things-when-applying-ci-cd/>
- <https://www.muycomputerpro.com/2020/01/29/continuous-integration-continuous-delivery-casi-todo-lo-que-tienes-que-saber>
- <https://www.sciencedirect.com/science/article/pii/S0164121217300353>
- <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>
- <https://continuousdelivery.com/>
- <https://www.informit.com/articles/article.aspx?p=1621865>
- <https://aws.amazon.com/es/devops/continuous-delivery/>



¡GRACIAS!

¿Alguna pregunta?

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**