

Puesta en Producción Segura

Unidad 0.

Ciclos de Desarrollo Software Seguros



"Integrar seguridad en DevOps no es frenar la velocidad del desarrollo, sino garantizar que cada entrega sea segura, estable y confiable."

Objetivos

- Comprender el concepto de ciclo de desarrollo de software (SDLC).
- Diferenciar los principales tipos y modelos de SDLC.
- Introducir el concepto de Integración Continua (CI) y Despliegue Continuo (CD).
- Entender la filosofía y prácticas de DevOps.
- Comprender la importancia de integrar la seguridad en el ciclo de desarrollo mediante SecDevOps.
- Aplicar un enfoque crítico hacia la seguridad en todo el ciclo de vida del software.



Contenidos

01 SDLC: Ciclo de desarrollo software

02 Tipos de SDLC

03 Seguridad en el SDLC

04 Integración continua y Entrega continua (CI/DC)

05 DevOps - SecDevOps



01

SDLC: Ciclo de Desarrollo Software



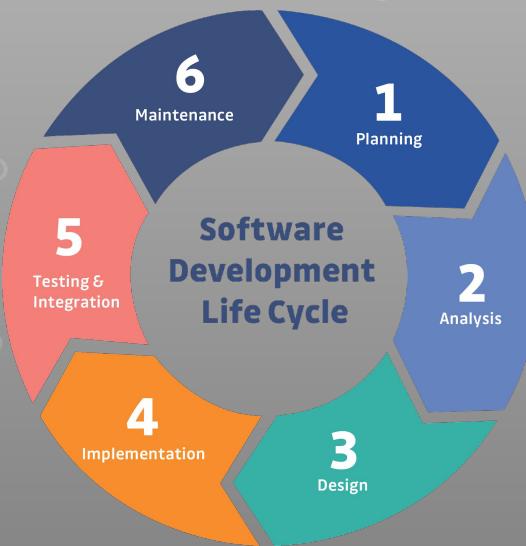
01. Seguridad en el ciclos de desarrollo Software

¿Qué es el SDLC?

El SDLC es el ciclo de vida de desarrollo software

Fases del SDLC

- Planificación
- Análisis
- Diseño
- Implementación
- Pruebas
- Integración
- Mantenimiento





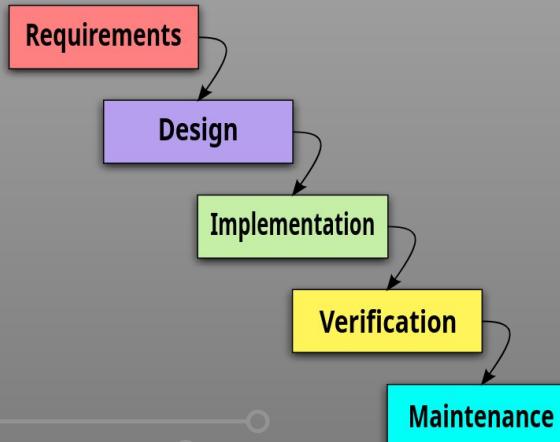
02

Tipos de SDLC

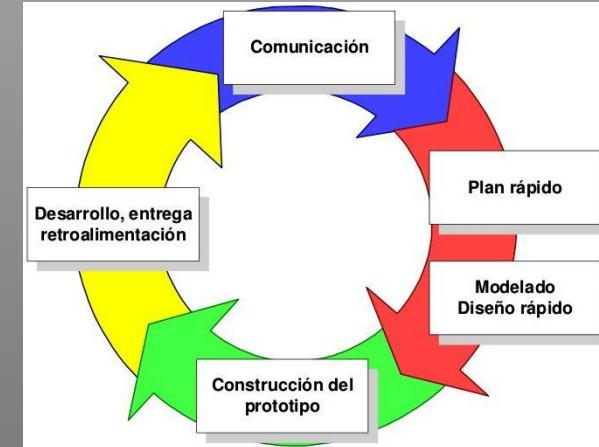
01. Modelos de SDLC: Ventajas e inconvenientes

¿Cómo se desarrollan estas fases?

- Módelo Clásico o en cascada



- Modelos en prototipos



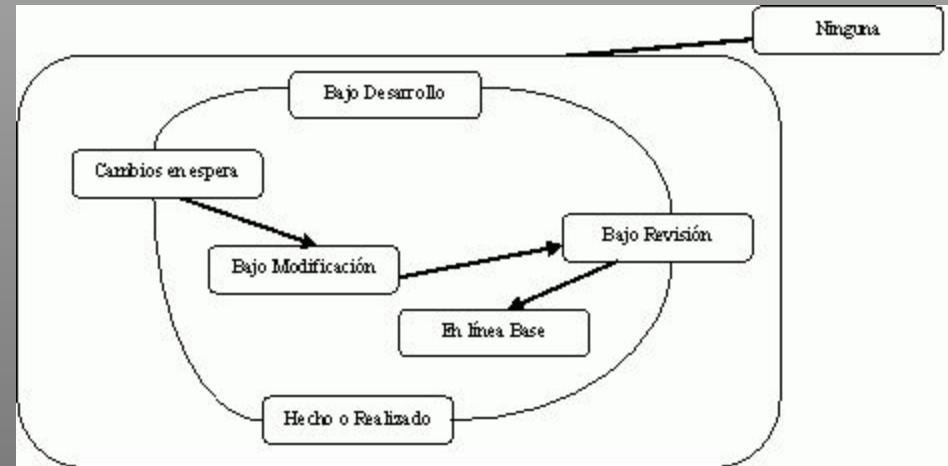
01. Seguridad en el ciclos de desarrollo Software

¿Cómo se desarrollan estas fases?

- Modelos en espiral



- Modelos de desarrollo concurrente



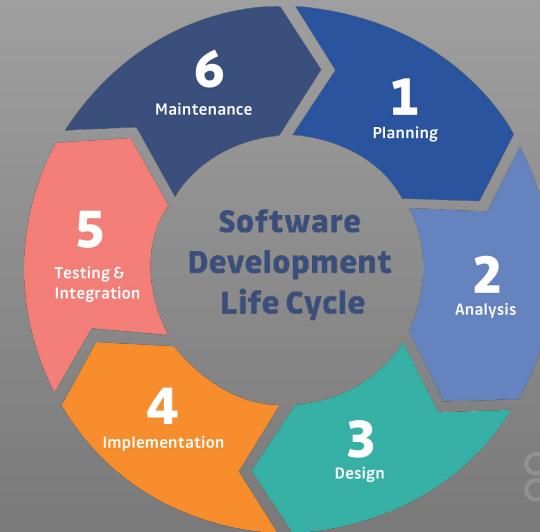
01. Seguridad en el ciclos de desarrollo Software

¿Cómo se desarrollan estas fases?

- Módelo de Desarrollo Unificado



- Metodologías Agiles





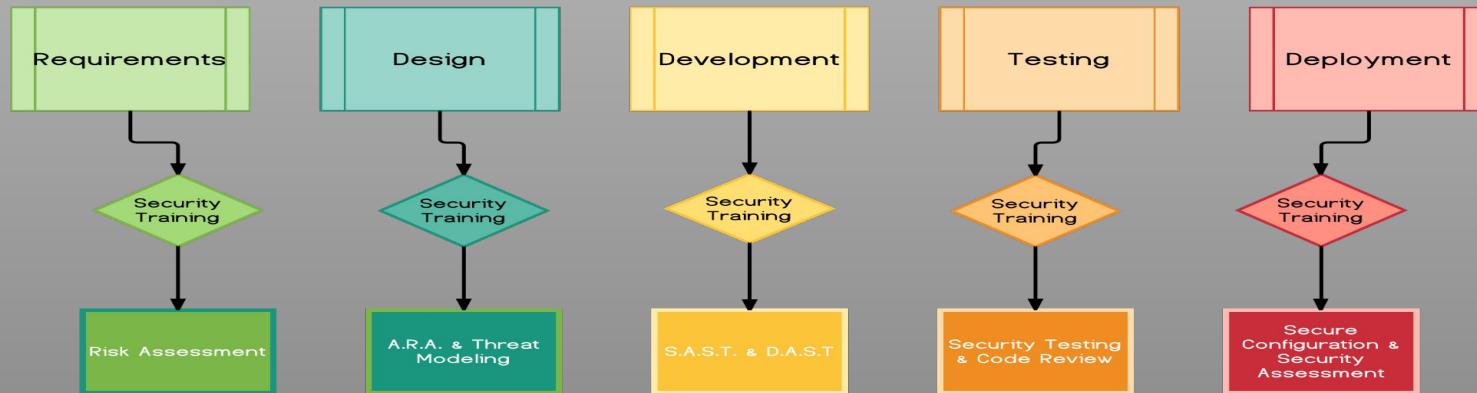
03

Seguridad en el SDLC



01. Seguridad en el ciclos de desarrollo Software

FROM SDLC TO S-SDLC



COST OF DEFECTS



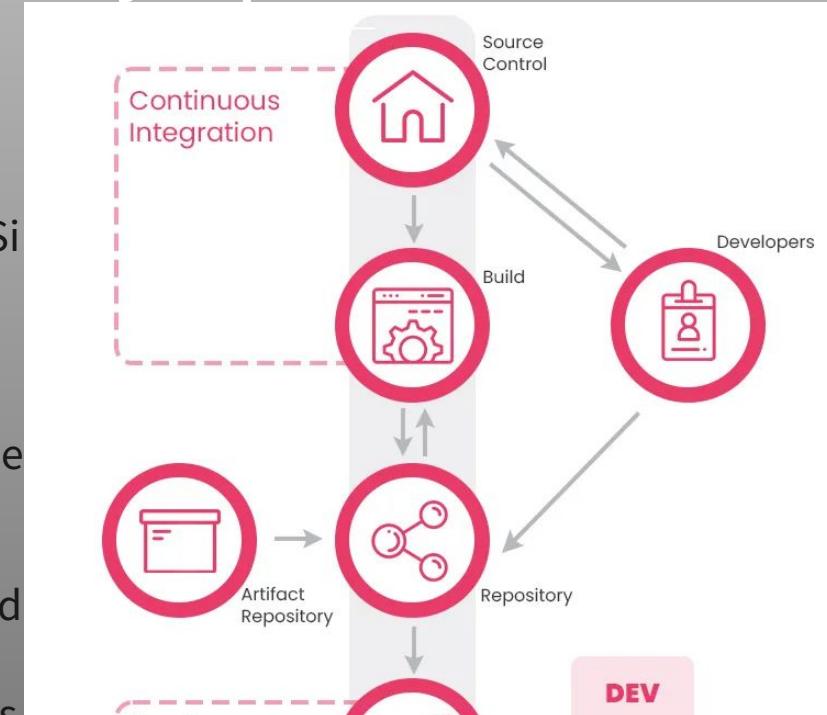
04

Integración contínua y Entrega contínua (CI/DC)



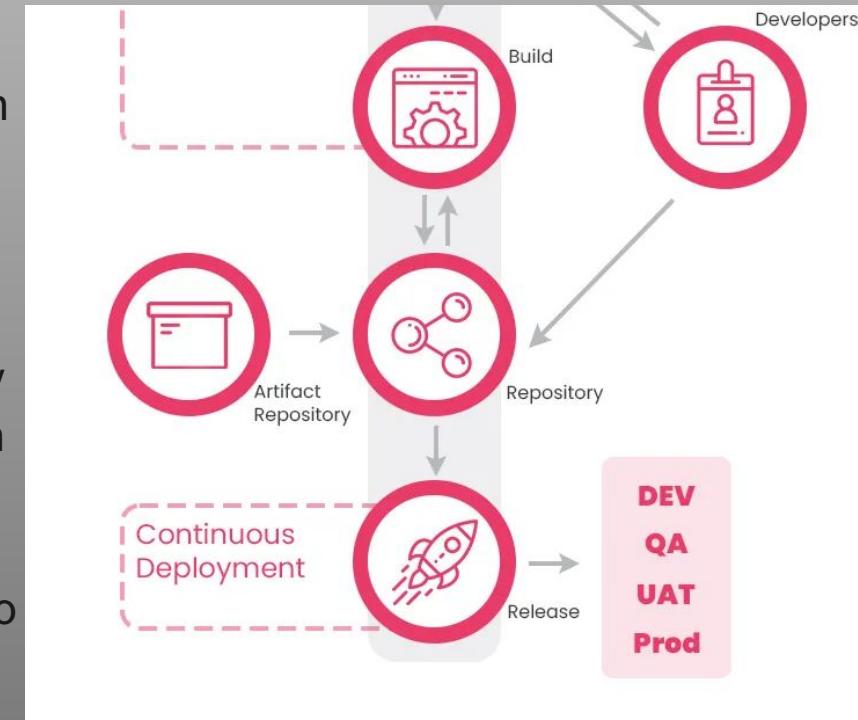
Integración del desarrollo y operación

- La **integración continua (CI)**, la **entrega continua (CD)** y las **pruebas continuas (CT)** son los tres principales métodos mediante los cuales muchos equipos de desarrollo adoptan una estrategia ágil. Si bien cada uno cumple un objetivo ligeramente diferente, cuando se combinan, pueden ayudar significativamente al equipo a alcanzar la velocidad y la calidad, dos de los objetivos más importantes de cualquier equipo de desarrollo.
- Las **ventajas** que aporta son: Reducir la complejidad de proyectos largos, Reducir los errores, Facilitar el despliegue, Entregar más rápido y Mejorar los costes y la productividad



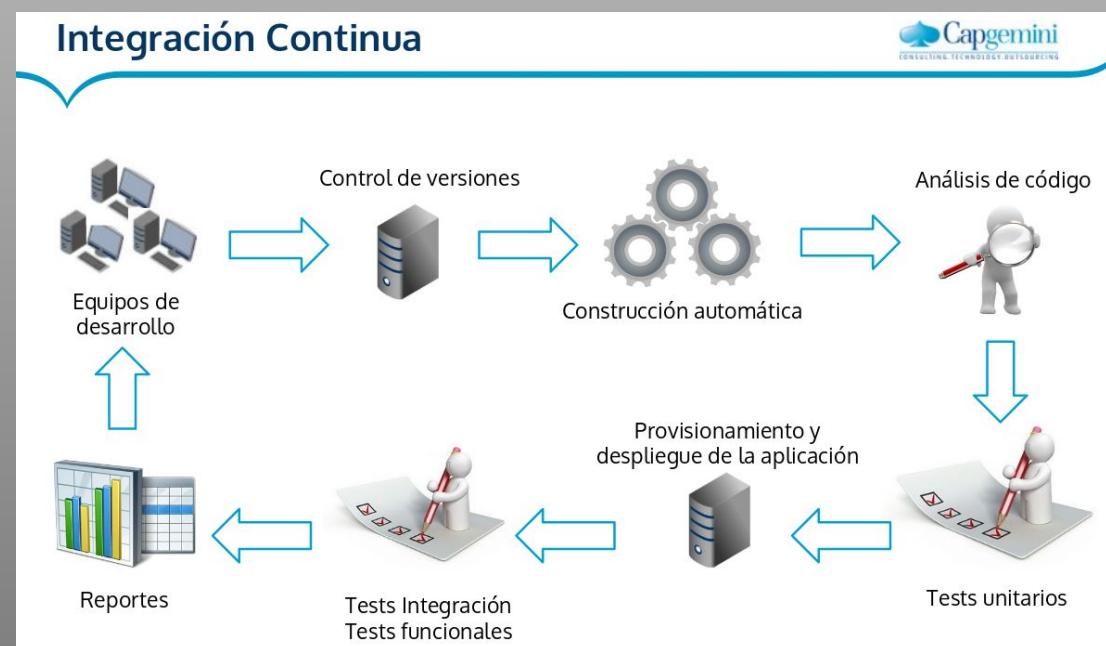
Integración del desarrollo y operación

- La **Integración Continua** («Continuous integration») se centra en la integración del trabajo de los desarrolladores individuales en un repositorio principal varias veces al día para detectar errores de integración de manera temprana y acelerar el desarrollo colaborativo.
- La **Distribución Continua** («Continuous delivery ») tiene que ver con la reducción de la fricción en el proceso de implementación o liberación, la automatización de los pasos necesarios para implementar una compilación para que el código se pueda liberar de forma segura en cualquier momento.



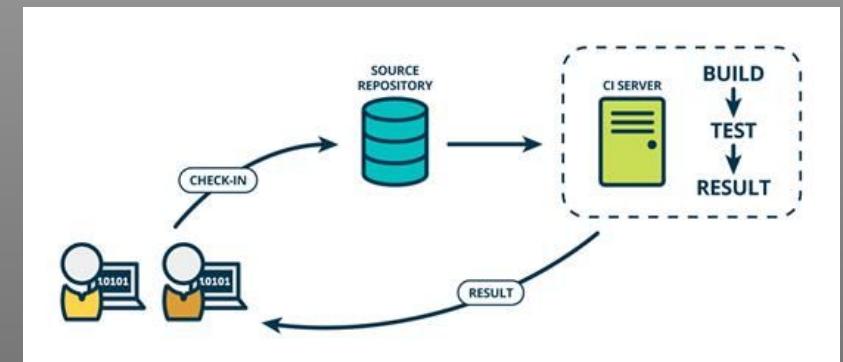
Integración del desarrollo y operación

- La **Implementación Continua** («Continuous deployment») va incluso un paso más allá al implementarse automáticamente cada vez que se realiza un cambio de código.
- Aparte de ellos también tenemos otros métodos cómo son CT (**Calidad Continua**) o IC **herramientas de automatización y procesos de DC**



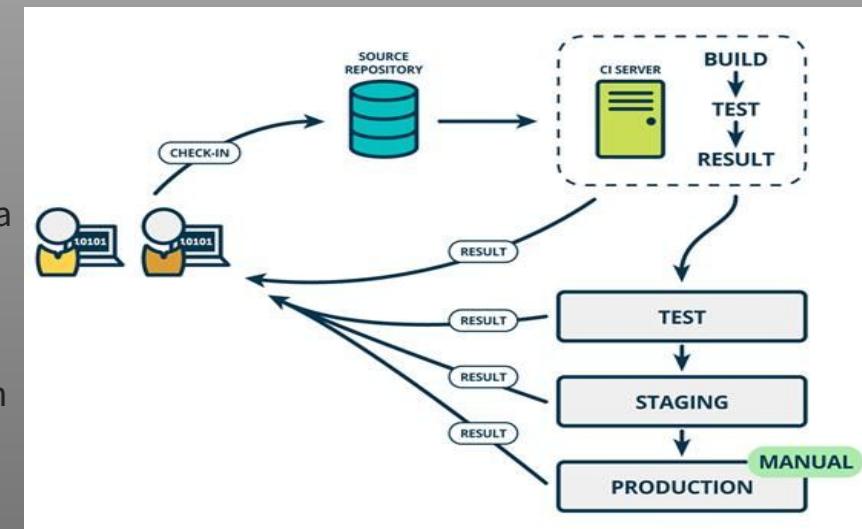
Integración continua (CI)

- La Integración Continua («Continuous integration») es una práctica que alienta a los desarrolladores a integrar su código en una rama principal de un repositorio compartido de forma temprana y frecuente. En lugar de crear características de forma aislada e integrarlas al final de un ciclo de desarrollo, cada desarrollador integra el código con el repositorio compartido varias veces a lo largo del día.
- La idea es minimizar el costo de la integración haciendo una prueba de integración temprana. Los desarrolladores pueden descubrir conflictos en los límites entre el código nuevo y del existente desde el principio, lo que hace que los conflictos aún sean relativamente fáciles de reconciliar. Una vez que se resuelve el conflicto, el trabajo puede continuar con la confianza de que el nuevo código respeta los requisitos de la base de código existente.
- Se basa en un conjunto de pruebas sólidas y un sistema automatizado para ejecutar esas pruebas. Cuando un desarrollador combina código en el repositorio principal, los procesos automatizados inician una compilación del nuevo código. Posteriormente, los entornos de prueba se ejecutan contra la nueva compilación para comprobar si se introdujeron problemas de integración. Si falla la fase de compilación o la fase de prueba, se alerta al equipo para que pueda trabajar en la reparación de la compilación.



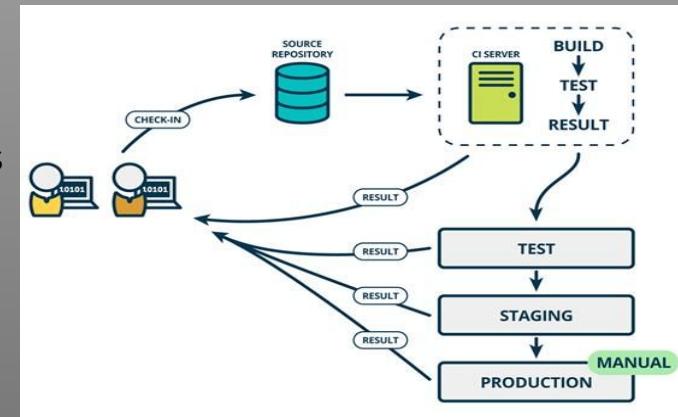
Distribución o Entrega continua (CD)

- La Distribución Continua es una extensión de la Integración Continua. Se centra en la automatización del proceso de entrega de software para que los equipos puedan implementar su código de forma fácil y confiable en cualquier momento. Al asegurar que la base de código siempre se encuentre en un estado desplegable, la distribución del software se convierte en un evento sin complicaciones y sin un ritual complicado. Los equipos pueden estar seguros de que pueden lanzar cuando lo necesiten, sin una coordinación compleja o pruebas de última etapa. Al igual que con la Integración Continua, la entrega continua es una práctica que requiere una combinación de mejoras técnicas y organizativas para ser efectiva.
- Desde el punto de visto tecnológico, la Distribución Continua se apoya fuertemente en los canales de despliegue para automatizar los procesos de prueba e implementación. Una canalización de implementación es un sistema automatizado que ejecuta conjuntos de prueba cada vez más rigurosos contra una compilación como una serie de etapas secuenciales. Esto se inicia cuando la Integración Continua ha finalizado una etapa, por lo que una configuración de Integración Continua confiable es un requisito previo para implementar la Distribución Continua.



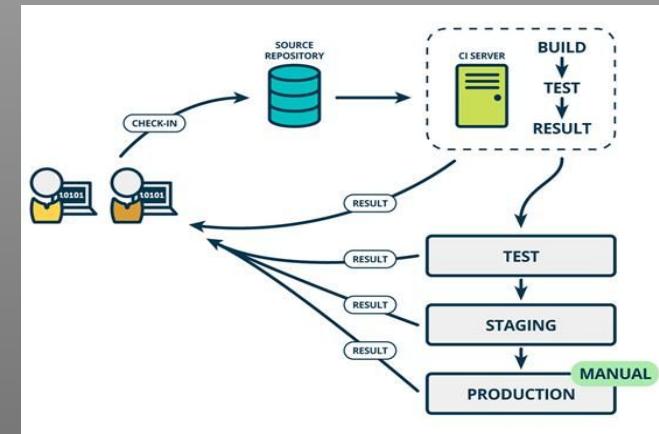
Implementación o Despliegue continuo (CI)

- La Implementación Continua es una extensión de la Distribución Continua e implementa automáticamente cada versión que pasa por el ciclo de prueba completo. En lugar de esperar a que un controlador de acceso humano (persona, N. del T.) decida qué y cuándo desplegarlo en la producción, un sistema de Implementación Continua instala todo lo que ha atravesado con éxito la canalización de implementación. Tenga en cuenta que, si bien el nuevo código se implementa automáticamente, existen técnicas para activar nuevas características en un momento posterior o para un subconjunto de usuarios. La implementación lleva automáticamente las características y soluciones a los clientes rápidamente, fomenta los cambios más pequeños con un alcance limitado y ayuda a evitar la confusión sobre lo que se implementa actualmente en la producción.
- La implementación continua también permite a las organizaciones beneficiarse de la retroalimentación temprana constante. Las características pueden ponerse inmediatamente a disposición de los usuarios y los defectos o las implementaciones inútiles se pueden detectar antes de que el equipo dedique un gran esfuerzo en una dirección improductiva. Recibir comentarios rápidos de que una característica no es útil permite al equipo cambiar de enfoque en lugar de sumergir más energía en un área con un impacto mínimo.



Implementación o Despliegue continuo (CI)

- La Implementación Continua es una extensión de la Distribución Continua e implementa automáticamente cada versión que pasa por el ciclo de prueba completo. En lugar de esperar a que un controlador de acceso humano (persona, N. del T.) decida qué y cuándo desplegarlo en la producción, un sistema de Implementación Continua instala todo lo que ha atravesado con éxito la canalización de implementación. Tenga en cuenta que, si bien el nuevo código se implementa automáticamente, existen técnicas para activar nuevas características en un momento posterior o para un subconjunto de usuarios. La implementación lleva automáticamente las características y soluciones a los clientes rápidamente, fomenta los cambios más pequeños con un alcance limitado y ayuda a evitar la confusión sobre lo que se implementa actualmente en la producción.
- La implementación continua también permite a las organizaciones beneficiarse de la retroalimentación temprana constante. Las características pueden ponerse inmediatamente a disposición de los usuarios y los defectos o las implementaciones inútiles se pueden detectar antes de que el equipo dedique un gran esfuerzo en una dirección improductiva. Recibir comentarios rápidos de que una característica no es útil permite al equipo cambiar de enfoque en lugar de sumergir más energía en un área con un impacto mínimo.



Calidad continua (CT)

- **CT**, que también puede denominarse **calidad continua**, es la práctica de integrar y automatizar las actividades de prueba en cada "commit". CT ayuda a los desarrolladores a utilizar su tiempo de manera más eficiente cuando intentan reparar un error de código que fue escrito hace años. Para corregir el error, los desarrolladores deberían primero recordarse a sí mismos de qué código era, deshacer cualquier código que se haya escrito sobre el código original, y luego volver a probar el nuevo código, no un proceso corto. Las pruebas que se realizan cada compromiso, cada pocas horas, todas las noches y semanalmente no sólo aumenta la confianza en la calidad de la aplicación, sino que también impulsa la eficiencia del equipo.

Conceptos y prácticas clave para los Procesos Continuos.

- **Cambios pequeños e iterativos:** Los desarrolladores deben practicar el dividir el trabajo más grande en pequeños pedazos y combinarlos al repositorio apenas estén listos , y luego continuamente a lo largo del desarrollo. De esta forma el costo de la integración disminuye.
- **Desarrollo basado en tronco:** El trabajo se realiza en la rama principal del repositorio o se fusiona en el repositorio compartido a intervalos frecuentes. Las ramas con característica de vida corta son permisibles siempre que representen pequeños cambios y se fusionen lo antes posible.
- **Mantener rápidas las fases de construcción y prueba:** Cada uno de los procesos se basa en construcciones (compilaciones, N. del T.) y pruebas automatizadas para validar la corrección. Debido a que los pasos de compilación y prueba deben realizarse con frecuencia, es esencial que estos procesos se simplifiquen para minimizar el tiempo dedicado a estos pasos.



05

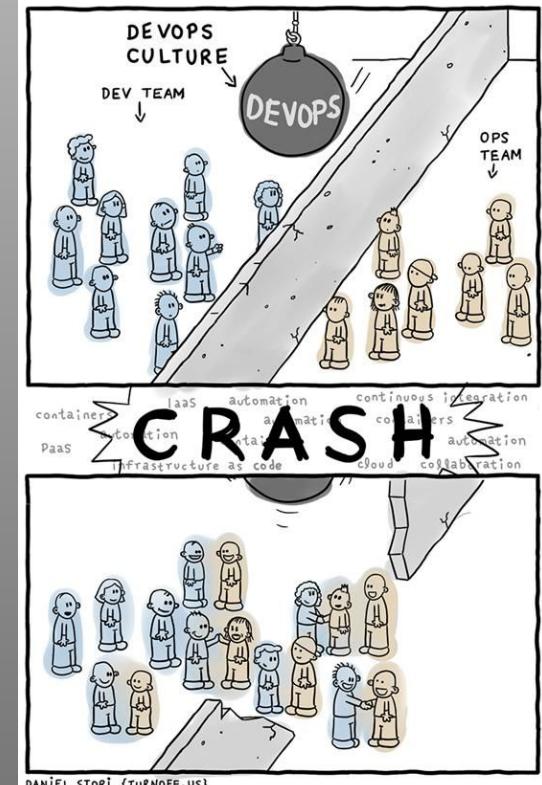
DevOps - SecDevOps



Integración del desarrollo y operación (DevOps)

DevOps (Developments & Operations) es una combinación de desarrollos y Operaciones que podríamos interpretarlo como **Técnicas de desarrollo ágil**.

DevOps permite que los roles que antes estaban aislados (desarrollo, operaciones de TI, ingeniería de la calidad y seguridad) se coordinen y colaboren para producir productos mejores y más confiables. Al adoptar una cultura de DevOps junto con prácticas y herramientas de DevOps, los equipos adquieren la capacidad de responder mejor a las necesidades de los clientes, aumentar la confianza en las aplicaciones que crean y alcanzar los objetivos empresariales en menos tiempo.



Integración del desarrollo y operación (DevOps)

- Es una metodología que es un conjunto de procesos, métodos y colectiva del sistema, que se utiliza para promover el desarrollo de aplicaciones, operación y mantenimiento de aplicaciones y control de calidad (QA) departamento. La comunicación, la colaboración y la integración se utilizan con el objetivo de romper las barreras y las diferencias entre el desarrollo y las operaciones tradicionales.
- DevOps es una cultura, el ejercicio o la práctica de potenciar a los desarrolladores de software (DEV) y de operación y mantenimiento de equipos técnicos (OPS), que permite la construcción, pruebas, software de liberación a través de la entrega de software automatizado y cambios arquitectónicos. Es más rápido, frecuente y fiable, concretamente se trata de mejorar la comunicación y colaboración durante la administración e implementación de software, cuyo objetivo es ser más rápido, más fiable y la elaboración de productos de mayor calidad.
- DevOps no es un tipo de software particular, sino un conjunto de procesos y métodos.

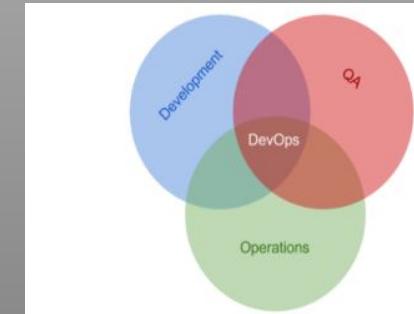


Figura 3.1: Estructura de los roles tradicionales frente a DevOps

Fases de DevOps



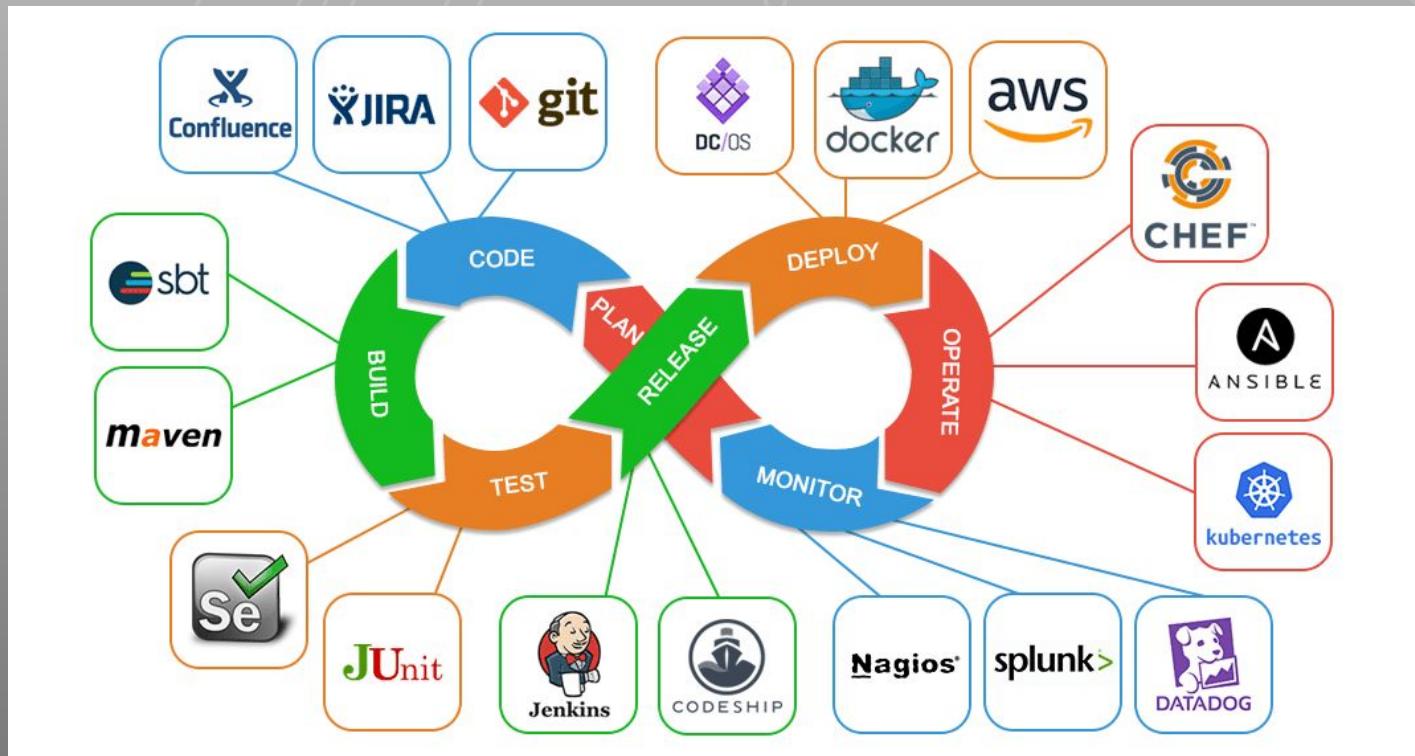
Fases del SDLC

- Planificación
- Construcción
- Pruebas
- Integración continua
- Despliegue continuo
- Operación
- Feedback continuo

DevSecOps

- **DevSecOps (desarrollo más seguridad más operaciones)**, es un aumento de DevOps para permitir que las prácticas de seguridad se integren en el enfoque de DevOps. El modelo tradicional de equipo de seguridad centralizado debe adoptar un modelo federado que permita a cada equipo de delivery tener en cuenta los controles de seguridad correctos en sus prácticas de DevOps.
- El objetivo principal de **DevSecOps** es automatizar, monitorear y aplicar la seguridad en todas las fases del ciclo de vida del software, es decir: planificar, desarrollar, construir, probar, lanzar, entregar, implementar, operar y monitorear. La aplicación de seguridad en cada etapa del proceso de desarrollo de software permite la integración continua, lo que reduce el costo de cumplimiento y entrega el software más rápidamente.
- **DevSecOps** significa que cada empleado y equipo es responsable de la seguridad desde el principio, y deben tomar decisiones de manera eficiente y ponerlas en acción sin perder la seguridad.

Herramientas en DevOps



Bibliografía y Webgrafía

- <https://www.ks7000.net.ve/2018/02/03/una-introduccion-la-integracion-entrega-e-implementacion-continua-software/>
- <https://www.redhat.com/es/topics/devops/what-is-devsecops>

Gracias!

¿Alguna pregunta?



informatica.iesvalledeljerteplasencia.es



coordinacion.cenfp@iesvp.es



C/ Pedro y Francisco González, s/n
10600, Plasencia (Cáceres)



927 01 77 74

