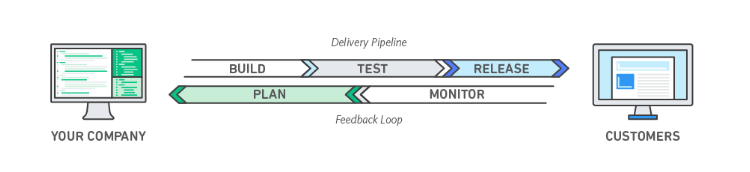
DevOps (Development and Operations):

DevOps combina el desarrollo (Dev) y las operaciones (Ops) para unir a personas, procesos y tecnología en el planeamiento de aplicaciones, el desarrollo, la entrega y las operaciones. DevOps permite a los roles previamente asilados, como el desarrollo, las operaciones de TI, la ingeniería de calidad y la coordinación y colaboración de seguridad mantenerse interconectados.

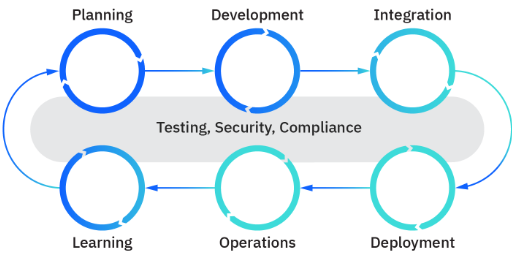
Los equipos adoptan la cultura, las prácticas y las herramientas de DevOps para aumentar la confianza en las aplicaciones que crean, responder mejor a las necesidades de los clientes y lograr objetivos empresariales más rápido. DevOps ayuda a los equipos a proporcionar valor continuamente a los clientes mediante la producción de productos mejores y más confiables.



DevOps y el ciclo de vida de la aplicación:

DevOps influye en el ciclo de vida de la aplicación a lo largo de su planeamiento, desarrollo, entrega y fases de operaciones.

Cada fase se basa en las demás y ninguna es específica de un rol en específico.



Fases de del estilo de vida de la aplicación DevOps:

* Planificación:

En este flujo de trabajo, los equipos determinan las nuevas características y funcionalidades del próximo lanzamiento, a partir de comentarios y casos de estudio de usuarios finales priorizados, así como contribuciones de todos los stakeholders internos

* Desarrollo:

Incluye todos los aspectos del desarrollo del código del software.

1º. Seleccionar un entorno de desarrollo.

2º. Escribir, probar, revisar e integrar el código.

3º. Compilar el código en máquinas virtuales para implementarlos en varios entornos.

4º. Usar el control de versiones (con Gilt que es un sistema de control de versiones distribuido) para colaborar en el código y trabajar en paralelo.

* Entrega:

La entrega es el proceso de implementación coherente y confiable de aplicaciones en entornos de producción, preferiblemente por entrega continua. (entrega continua es el proceso de automatizar la compilación, la prueba, la configuración y la implementación desde una compilación a un entorno de producción)

1º. Se de fine un proceso de administración de versiones con fases de aprobación manuales claras.

2º. Establecer puertas automatizadas para mover aplicaciones entre distintas versiones hasta la versión final a los clientes.

3º. Automatizar los procesos de entrega.

* Operations:

Implica mantener, supervisar y solucionar problemas de aplicaciones en entornos de producción.

* Aprendizaje (a veces llamado retroalimentación continua):

Esta es la recopilación de comentarios de usuarios finales y clientes acerca de las características, la funcionalidad, el rendimiento y el valor empresarial para volver a la etapa de planificación y añadir tanto mejoras como nuevas características al próximo lanzamiento.

* Pruebas continuas:

Los ciclos de vida tradicionales de DevOps incluyen una etapa de "prueba" discreta que se produce entre la integración y la implementación.

* Seguridad:

Mientras que las metodologías en cascada y las implementaciones ágiles "añaden" los flujos de trabajo de seguridad después de la entrega o la implementación, DevOps se esfuerza por incorporar la seguridad desde el principio (planificación), cuando los problemas de seguridad son más fáciles y menos costosos de abordar, y continuamente durante el resto del ciclo de desarrollo.

* Conformidad:

El cumplimiento normativo (gestión y riesgo) también se aborda mejor al principio y durante todo el ciclo de vida del desarrollo.

Cultura Devops:

Generalmente se admite que los métodos DevOps no pueden funcionar sin un compromiso con la cultura DevOps, la que se puede resumir como un enfoque organizativo y técnico diferente al desarrollo de software.

A nivel de organización, DevOps requiere comunicación continua, colaboración y responsabilidad compartida entre todos los involucrados en la entrega de software, desarrollo de software y equipos de operaciones de TI.

En la mayoría de los casos, la mejor manera de lograrlo es descomponer estos silos y reorganizarlos en equipos de DevOps autónomos e interfuncionales que pueden trabajar en proyectos de código de principio a fin.

Herramientas DevOps:

* Herramientas de gestión de proyectos:

Herramientas que permiten a los equipos crear una lista de casos de usuarios (requisitos) que forman proyectos de codificación, desglosarlos en tareas más pequeñas y realizar un seguimiento de estas hasta su finalización.

* Repositorios de código fuente de colaboración:

Entornos de codificación controlados por ediciones que permiten a varios desarrolladores trabajar en la misma base de código.

* Líneas de trabajo CI/CD:

Herramientas que automatizan la extracción, la creación, la prueba y la implementación de código

* Marcos de automatización de pruebas: I

Incluyen herramientas de software, bibliotecas y mejores prácticas para automatizar pruebas de unidad, contrato, funcionalidad, rendimiento, usabilidad, penetración y seguridad.