

NUEVAS METODOLOGÍAS ÁGILES DE DESARROLLO Y NUEVOS PERFILES LABORALES

Recursos:

- Devops en menos de 3 minutos: <https://www.youtube.com/watch?reload=9&v=p-bOnV8FRMQ>
- [Integración continua, entrega continua y despliegue continuo. Ciclo de vida del SW](#)

1. El origen de DevOps

La actualización y puesta en producción de un producto hoy en día no puede llevar meses como antaño, y menos aún en el área de las aplicaciones web. Los administradores de sistemas y los desarrolladores están obligados a entenderse, a conocer el trabajo que el otro hace. Por ello se requiere un **desarrollo ágil**, lo cual explicaremos más adelante.

Uno de los objetivos de DEVOPS es reducir los tiempos de cada ciclo de desarrollo, es decir, reducir el tiempo necesario para que cada uno de los requisitos solicitados por los usuarios finales llegue a sus manos.

Para que esto pueda ocurrir tiene que pasar por todas las etapas del ciclo de vida de la aplicación, es decir, debe realizar el análisis y diseño de la solución, se debe codificar y probar de forma aislada, se tiene que integrar con el resto de modificaciones, se tiene que implantar en los distintos entornos para que se realicen las pruebas oportunas hasta que finalmente es validado y liberado a Producción.

Para reducir el tiempo necesario en cada uno de estos pasos, DEVOPS propone la máxima automatización de cada uno de ellos, de forma que la intervención humana sea mínima y cuando realmente sea necesaria. En este sentido hay que destacar las técnicas de desarrollo en paralelo, integración continua y entrega continua y despliegue continuo.

2. Desarrollo ágil

El desarrollo mediante mecanismos clásicos de desarrollo en el modelo de cascada deja de ser eficiente.

En el modelo tradicional como en el de cascada podía ocurrir que entre la versión 2.0 de un producto y la publicación de la 3.0 podían pasar u años. El desarrollo se produce durante ese largo periodo de tiempo y llega un momento en que se aplica el cambio de forma drástica y brusca, normalmente con una parada de servicio.

Eso en una aplicación web no se puede plantear. A diferencia se hacen pequeños cambios, secuenciales, iterativos, en los que se van incluyendo las nuevas una funciones de forma continua y no existe versión 2 ni existe versión 3, **existe siempre una versión en constante cambio.** Este mecanismo permite también, estando en un paso intermedio, añadir una funcionalidad que no se había pensado inicialmente porque no se tenía esa necesidad.

Este continuo cambio hace que la programación también se adapte mejor a los cambios, y se adapten mejor al tipo de demanda de aplicación que están teniendo los usuarios.

Hoy en día el desarrollo ágil es considerado como buenas prácticas, de forma general y las buenas prácticas en desarrollo. Además incluye una variedad importante de metodologías,, muy adecuadas para aplicaciones web.

Por otro lado hace que ese mecanismo sea incompatible con las formas clásicas de administrar los sistemas. **Los desarrolladores desarrollan, pero quién pone los cambios en producción son los administradores de sistemas,** por lo que si el administrador de sistemas tienen una forma de trabajar y los desarrolladores tienen otra. Ahí se produce una fricción, tenemos un problema, porque unos están cambiando continuamente y los otros siguen trabajando como siempre, es decir, si hay algún cambio hay una parada, pero no se puede parar porque se trata de una aplicación web.

Los sysadmin y los desarrolladores están condenados a entenderse y escucharse.

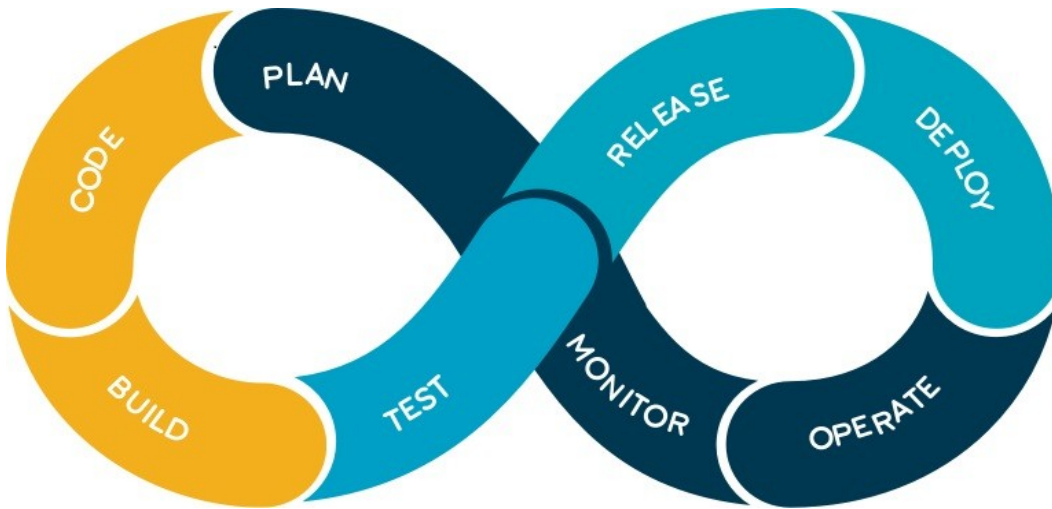
Los desarrolladores tienen que asumir que es muy importante desarrollar de una manera que no ponga en peligro la estabilidad y la seguridad de los sistemas, y por otro lado los sysadmin **tienen que asumir que es necesario las modificaciones que se hagan en la aplicaciones**ve irlas incorporando de manera adecuada, a un ritmo similar o en sintonía con el que se van desarrollando, es decir, el personal de sistemas tiene que cambiar la forma de trabajar continuamente.

Se empiezan a producir las necesidades de incorporar cambios en la parte de sistema. En la parte de desarrollo se incluyeron ya esos cambios en su momento, y hoy en día es una práctica absolutamente generalizada, aunque hay mucha gente que sigue desarrollando de forma tradicional.

Un ejemplo de esto sería el tema de la integración continua, un tipo de práctica que se hace en desarrollo ágil. Consiste en que **se aplican los cambios de forma continua, todos los días, es decir, los desarrolladores están continuamente trabajando y esos cambios se aplican, con idea de que sean cambios definitivos.** Lo que se hace es que se automatiza la integración de la compilación, en el caso de que la haya, y la realización de pruebas para verificar que esos cambios son correctos. Esto se hace con el objetivo de detectar pronto un error y si lo hay, volverlo a la fase de desarrollo para volverlo a modificar.

Todo esto implica nuevas metodologías de desarrollo más ágiles y nuevos perfiles profesionales como el de **Devops**, **QA Tester (Quality Assurance)**, etc.

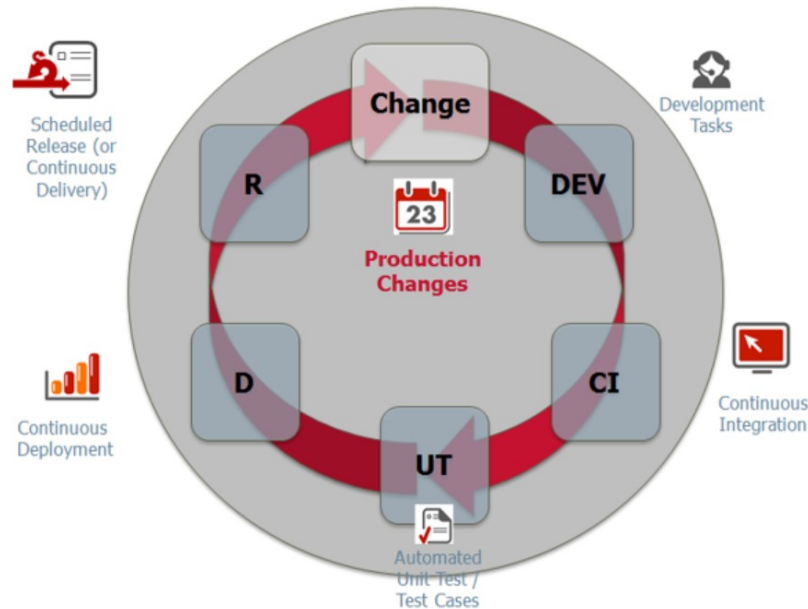
Aparecen los conceptos de Integración continua, entrega continua y despliegue continuo.



3. INTEGRACIÓN CONTINUA (CI), ENTREGA CONTINUA (CD) Y DESPLIEGUE CONTINUO (CD)

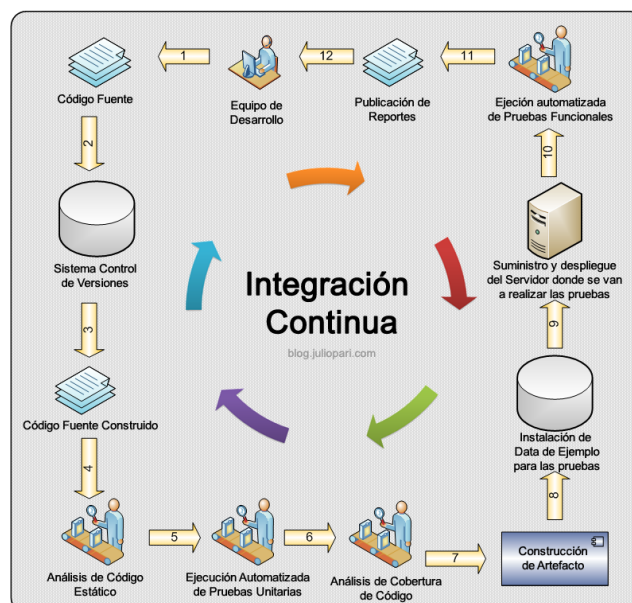
El **ciclo de vida** de desarrollo SW (ver video OpenWebinars):

1. Integración continua
2. Entrega continua
3. Despliegue continuo: puesta en producción (de forma automática)



Integración continua

- Se descarga del control de versiones las últimas fuentes con los cambios implementados a un directorio de trabajo.
- Se realiza la compilación de estas fuentes.
- Se lanza el análisis estático y de cobertura de las fuentes.
- Se instala en el entorno (compartido) de Desarrollo.
- Se lanzan las pruebas automáticas.
- Se generan los informes.



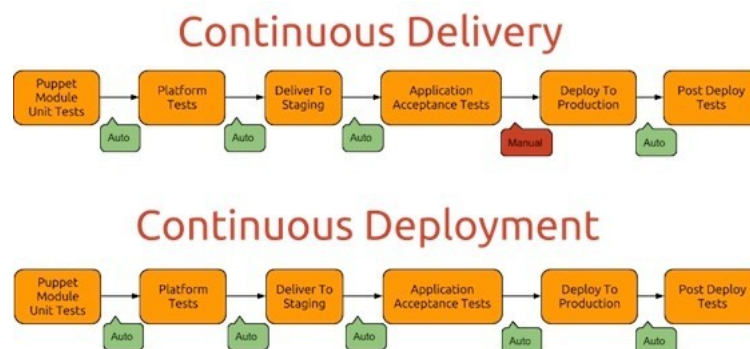
Entrega continua y despliegue continuo

El objetivo es facilitar a los usuarios finales lo antes posible las modificaciones que ha realizado desarrollo en la aplicación. Por lo que la entrega continua y el despliegue continuo pueden interpretarse como aplicar el concepto de integración continua al despliegue de aplicaciones.

Es importante tener en cuenta, que **la entrega continua** no consiste en desplegar en Producción cada cambio lo antes posible, sino que **cada cambio debe de estar disponible para ser desplegado en cualquier momento**.

Comienza una vez que la aplicación está construida y hay que implantarla en los siguientes entornos del ciclo de vida. Por ejemplo, entregar la nueva versión de la aplicación al equipo de testers para QA (Quality Assurance) para que pueda realizar las pruebas y finalmente entregarla a Operaciones (administradores de sistemas) para que pueda hacer su implantación en Producción.

La diferencia principal entre la entrega continua y el despliegue continuo es que el primero requiere de una aprobación manual antes de implantar en Producción, mientras que el segundo incluso la puesta en producción se realiza de forma automática una vez que se cubren todos los criterios definidos para la entrada en Producción para una aplicación.



La aparición del Cloud Computing: arquitectura en la nube con AWS y Azure y Git.

La utilización de infraestructura variable la suple claramente una tecnología que se conoce como Cloud Computing, que además hace muchas otras cosas, pero nos centraremos sólo en una, que incluye lo que se llama **la característica de elasticidad**, que permite que los recursos de infraestructura que esté utilizando una aplicación varíen de una forma muy rápida, casi instantánea, con el tiempo.

En el uso de Cloud Computing de nuevo se están utilizando servidores Linux en remoto. De las dos posibilidades que se tenían, la que se adaptó mejor a la utilización de Cloud Computing, algo que hoy está totalmente generalizado, fueron los sistemas Linux. Esto es así por lo que comentamos en el vídeo anterior por su dinamismo, por su facilidad para adaptarse a los cambios, por la versatilidad que tiene.

Esto hizo que en determinado momento se desarrollaran una serie de herramientas que encajaban muy bien en Cloud Computing y **se desarrollaran con software libre muchas de ellas**, otras no, para trabajar en entornos Linux.

Realmente hoy en día en Cloud Computing, incluso la solución de Cloud Computing de Microsoft, que es Azure, utiliza en muchísimos casos servidores Linux.