

# Es necesaria la integración continua

Vivimos en un mundo donde hoy en día cada vez se utilizan servicios y productos más enfocados en software. Pero antes de empezar a ampliar en el tema primeramente entenderemos que es integración continua (CI), según Martin Fowler<sup>1</sup> es una práctica de desarrollo donde los miembros del equipo integran su trabajo frecuentemente cada persona lo hace por lo menos una vez al día esta integración es verificada por el conocido "automated build" el cual detecta los errores más rápidamente.

En definitiva la aplicación de CI engloba todo el ciclo de vida del producto las modificaciones posibles que este se enfrentará, a agregarle nuevos features a este y otras más cosas que el proyecto pueda incluir gracias a esta grandiosa herramienta de la CI podemos automatizar muchas tareas de un proyecto e incluso ahorrarnos tiempo un ejemplo claro sería suponiendo que se tiene una versión que se desea liberar pero perderíamos tiempo en hacer todas las pruebas tanto funcionales y no funcionales ya que se trabajaría que un equipo desarrolla, otro equipo hace las pruebas funcionales, otro las no funcionales entonces ahora con la integración continua únicamente en el build se añaden estas funciones y el servidor de Integración comunicado con el repositorio donde se tiene el proyecto automatiza la pruebas, ejecuta pruebas de aceptación, métricas de calidad de código, compilación de componentes etc.

Por el momento mencionare varias prácticas a aplicar en la CI:

- **Mantener un único repositorio**

Con esto me refiero a que en un proyecto de software involucra muchos archivos que se necesitan juntos para así generar un producto, darle seguimiento a estos y mayormente si muchas personas están involucradas entonces se vio la necesidad de usar un sistema de control de versiones el cual debe contener todo lo requerido del proyecto, lo bueno del uso de SCV es que este nos permite crear múltiples ramas para así cada persona que esté involucrada en el proyecto puede manejar diferentes partes de este sin que otras estén afectadas y al final del día se integra todo lo de las ramas modificadas a la rama principal llamada "mainline".



<sup>1</sup> <http://www.martinfowler.com/aboutMe.html>

- **Automatizar la construcción del proyecto**

Muchas veces el movimiento de archivos de un lado a otro, los esquemas de DB hace complicada la compilación de un proyecto y pedirles a las personas que ejecuten manualmente las aplicaciones es una pérdida de tiempo entonces un build analiza los cambios se necesitan realizar como parte del proceso entonces la principal función del build es construir una aplicación únicamente con el código sin intervención manual, la regla de oro según Fowler es que el sistema debe ser capaz de ejecutarlo y correrlo en una maquina virgen con los archivos que se tienen en el repositorio con un único comando y teniendo corréndolo en su máquina.

- **Hacer que el build ejecute sus propios tests**

Tradicionalmente la compilación, ejecución y todas las cosas requeridas para que el programa sea ejecutado debía ser realizado por terceros una buena manera de obtener los bugs más rápidamente y de manera eficiente es incluir pruebas automatizadas por el proceso de construcción realizando TDD, BDD, test de aceptación para la realización de estos es necesario un test automático que chequee una gran parte del código base estos deben ser ejecutados por un simple comando y un propio chequeo, al ejecutar estos deben indicar si las pruebas fallaron o fueron exitosas.

- **Todos realizan Commits al "Mainline" todos los días**

La base principal de la integración es la comunicación, la integración permite a los desarrolladores comunicarse a otros acerca de los cambios que estos han realizado incluyendo las pruebas mencionadas anteriormente, la comunicación frecuente permite a las personas conocer los cambios de manera más rápida.

Entonces el desarrollador al realizar un commit actualiza su trabajo con la copia en el mainline resolviendo los conflictos de este. La regla general es que cada desarrollador debe hacer el commit al repositorio cada día para así resolver los conflictos que se encuentren más rápidamente.

- **Cada commit debe construir el mainline en la máquina de integración**

Normalmente las personas no realizan una actualización o construcción antes de realizar el commit o a veces las ambientes de las máquinas de los desarrolladores son diferentes como resultado se debe asegurar que los builds se realizan regularmente en la máquina de integración y si este es exitoso entonces se debería considerar el commit realizado. El desarrollador que realice el commit debe estar monitoreando el mainline para arreglar errores.

- **Arreglar Builds rotos inmediatamente**

Cuando el build mainline está roto es decir no funcional debido a que los desarrolladores no han sido lo suficientemente cuidadosos al realizar las actualizaciones y las construcciones localmente antes de realizar el commit lo más importante es repararlos lo más rápido posible para que así no perjudicar a los demás desarrolladores que estén involucrados en la misma rama o en si afectando todo el proyecto. No se refiere a que todo el equipo debe trabajar en el arreglo y detener todo los que están haciendo usualmente se realizan por pocas personas sabiendo que la priorización del arreglo del build es urgente.

- **Mantener un build rápido**

El principal punto de CI es de darle a los desarrolladores una retroalimentación rápida ocasionalmente los builds que se tardan a veces hasta 1 hora son totalmente irrazonables, por lo tanto la guía principal es que este tarde 10 min estará en lo razonable. Por lo tanto merece un esfuerzo que este se pueda realizar ya que cada minuto que se logre disminuir en el build es un tiempo que disminuye para los desarrolladores realizar un commit.

- **Todos pueden ver los que sucede**

La CI se asegura que todos los involucrados en el proyecto puedan fácilmente ver el estado del sistema y los cambios que se han realizado. Una de las cosas más importantes que se deben realizar es que este notifique a los desarrolladores el progreso en cada momento. Un ejemplo puede ser las notificaciones que Jenkins manda a los correos de los desarrolladores involucrados en el proyecto para que estos puedan saber las modificaciones que se realizó a este.

- **Automatizar el despliegue**

Para realizar la CI se precisa de un entorno, pruebas, producción ya que se debe tener este proyecto en varios entornos varias veces por lo tanto es más óptimo realizarlo de una manera automática para poder hacerlo de una manera más fácil.

En conclusión podemos decir que CI nos ayuda a:

- la reducción de riesgos y tiempo ya que al realizar integraciones continuas (diariamente) del proyecto se pueden mitigar los riesgos y evitar el tiempo que se pasaría arreglando errores si se realizara una integración cada 2 o 3 meses ya que este incluiría una cantidad vasta de errores en la cual se debería trabajar por mucho tiempo.
- También otro punto en el cual tenemos una ventaja muy grande es el proceso repetitivo y la duplicación de código ya que un proceso repetitivo manualmente se puede volver a realizar ya que las personas nos equivocamos fácilmente y realizarlo y podríamos realizarlo doblemente en cuanto a la duplicación de código por lo que se trabaja en varias ramas y solo se actualizan las copias que se encuentran el mainline se evita que varias personas del equipo trabajen en la misma característica varias veces por
- otro lado podemos tener una visibilidad más amplia del estado del proyecto en el que trabajamos por la automatización de compilación, testeo, métricas de calidad entre otras ya que el análisis de esto nos permite tener estadísticas de cómo se va desarrollando el estado.
- Se logra la comunicación y organización tanto del equipo para el desarrollo del software y las actualizaciones de este ya que debido a eso los desarrolladores se comunican entre sí para un desarrollo más óptimo.
- Y por último la posibilidad de disponer en el servidor demostraciones de la aplicación para que así los clientes puedan comprobar los cambios día a día.

En cuanto a servidores del CI y CV podemos encontrar varios en el mercado tanto como pago como opensource, que se acomoden a las expectativas de cada empresa.