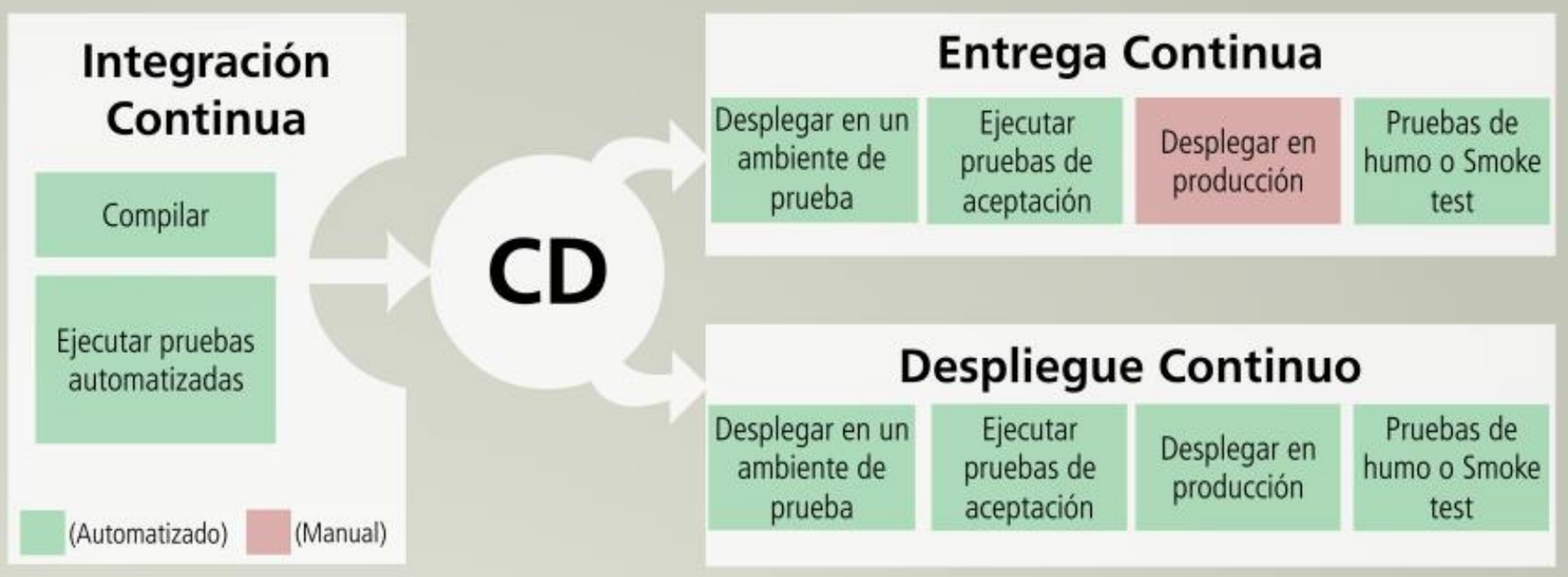
Continuous Integration Delivery Deployment

Introducción

Continuous Integration, C. Delivery y C. Deployment
Tres prácticas conceptualmente relacionadas, no solo porque se basan en los mismos dos pilares (aumento en la frecuencia y automatización), sino porque tanto C. Delivery como C. Deployment se tratan de una extensión de C. Integration siendo alternativo uno del otro.

Abordaremos este tema estableciendo la relación entre los tres conceptos a fin de entender su importancia, ventajas y desventajas de implementar dichas prácticas.



Discusión

¿Continuous Delivery o Continuous Deployment?

Nuestro objetivo final debe ser siempre lograr un despliegue continuo, pero puede ser muy complejo de conseguir si se trata de un proyecto que actualmente está funcionando en un ambiente de producción con clientes y usuarios finales.

Debemos comenzar implementando una integración continua con pruebas unitarias básicas, no debemos iniciar focalizándonos en pruebas complejas, en lugar de esto intentemos automatizar el despliegue tan pronto como sea posible y llegar a una etapa en la que los despliegues en nuestro ambiente de prueba se realicen automáticamente.
La razón es que al tener despliegues automatizados, podremos concentrar nuestras energías en mejorar pruebas en lugar de tener que detener todo periódicamente para coordinar una versión.

Conclusión

La implementación de estas buenas prácticas conlleva un cambio en la mentalidad y la forma de trabajo del equipo y exige compromiso y dedicación. Pero al analizar las características, ventajas y desventajas entendemos que este esfuerzo vale la pena, el hecho reducir costos, agilizar tiempos y disminuir complejidad y riesgos permite concentrarnos en lo que realmente importa, la calidad del producto.

Con este concepto en mente nuestro fin debería ser liberar software a diario, buscando una implementación continua, pero asegurándonos que el resto de nuestra organización y que nuestros clientes están listos para afrontar este ritmo de trabajo.

Bibliografía

Continuous integration vs. continuous delivery vs. continuous deployment - Sten Pittet

Continuous Integration - Martin Fowler

Cómo diseñé mi póster científico de la A a la Z - neoscientia

Autores

Cahuana, Keyssi
Casares Diaz, Mauricio
Ludueña, Joaquín
Pinchiroli Biani, Santiago
Ribero, Martín

Integración Continua

Frecuencia:

Los desarrolladores que practican la Integración Continua mezclan sus cambios (merge) con la rama principal tan a menudo como sea posible, incluso varias veces al día.

Automatización:

Se requiere de un servidor de integración que no solo nos permita automatizar la compilación del código sino también configurar una batería de pruebas que se correrán contra dicha compilación, si todas las pruebas resultan exitosas el código se integrará a la rama principal, en caso contrario se generarán alertas indicando el problema.

Ventajas:

Facilita la liberación ya que todos los detalles de integración se resuelven con anticipación.

Los desarrolladores reciben alertas tan pronto como rompen la compilación y pueden trabajar para solucionarlo antes de pasar a otra tarea.

Menor cantidad de errores llegan a producción.

Los costos asociados a las pruebas se reducen drásticamente debido a que el servidor puede ejecutar cientos de pruebas en cuestión de segundos.

Desventajas:

Se requiere de un servidor de integración.

Debemos programar las pruebas automáticas para cada nueva feature.

Entrega Continua

Se basa en liberar al cliente con la mayor frecuencia posible. Es una extensión de la integración continua, una vez que el nuevo código integrado pasa las pruebas automatizadas se despliega automáticamente en un ambiente de prueba y se ejecutan las correspondientes pruebas de aceptación.

Ventajas:

Baja la complejidad de desplegar al ambiente de prueba.

Permite liberar al cliente con mayor frecuencia, acelerando así el ciclo de retroalimentación.

Disminuye la presión sobre las decisiones para pequeños cambios, se fomentan iteraciones más rápidas.

Desventajas:

Necesariamente debemos implementar integración continua correctamente ya que la misma proporciona la base para esta práctica.

Se deben implementar "Feature Flags" para probar features incompletas.

El despliegue al ambiente de producción se sigue llevando a cabo de forma completamente manual.

Despliegue Continuo

El despliegue continuo va más allá que la entrega continua, ya que si el nuevo código pasa todas las pruebas automatizadas en el ambiente de test, entonces se despliega automáticamente al ambiente de producción.

Ventajas:

Aumenta la velocidad de desarrollo, debido a que no hay necesidad de pausar el desarrollo para gestionar las versiones.

Las liberaciones son menos riesgosas, ya que se libera paulatinamente.

Los clientes ven un flujo continuo de mejoras y la calidad aumenta cada día, en lugar de cada mes, trimestre o año.

Desventajas:

La calidad del conjunto de pruebas automáticas determinará la calidad del producto liberado.

La documentación debe mantenerse necesariamente actualizada al ritmo de los despliegues, lo cual supone una dificultad.