

# Unidad 4 – Desarrollo bajo SCM.

## 1. Integración Continua

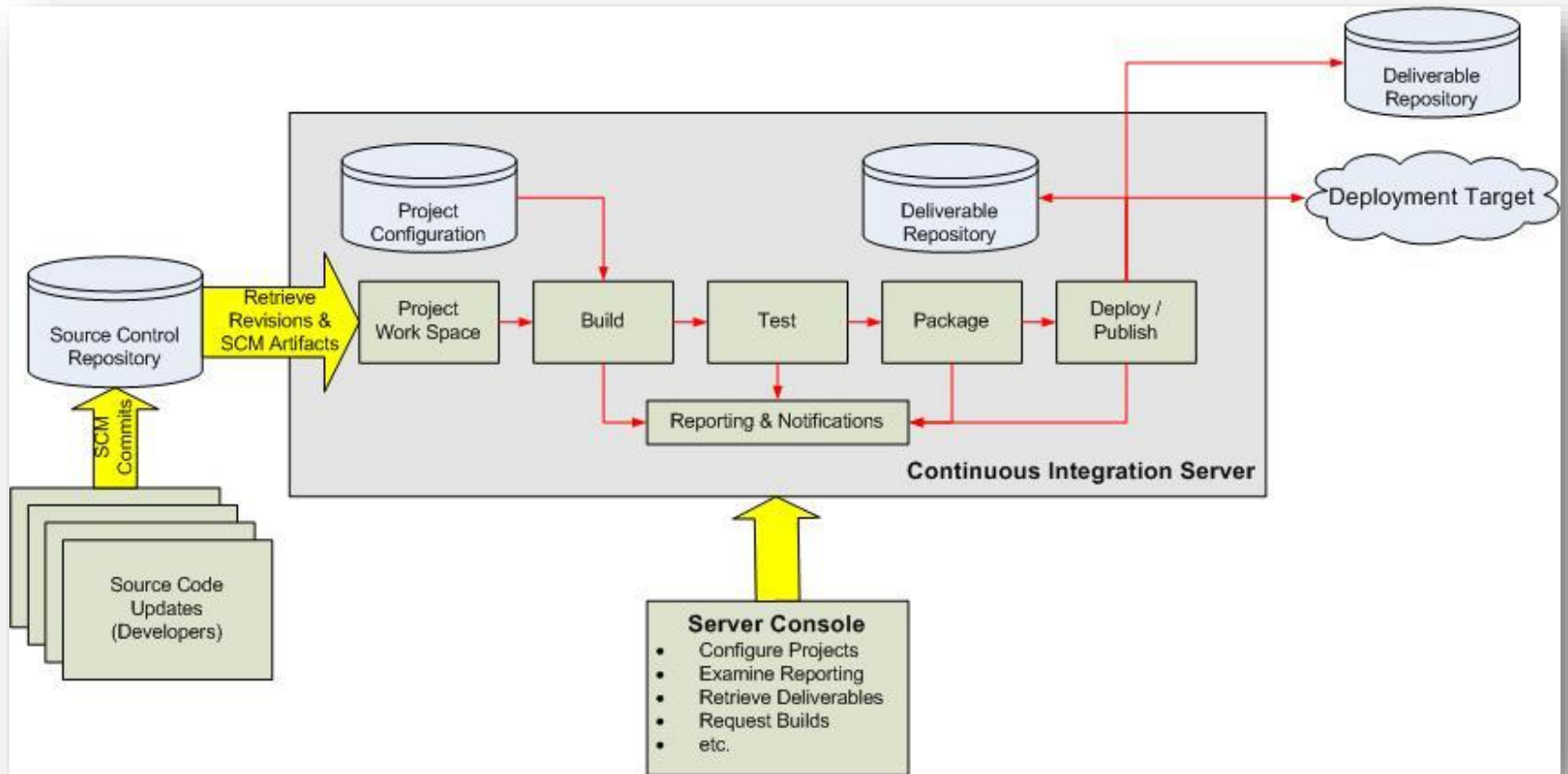
# Integración continua

- Práctica donde todos los **miembros** de un equipo **integran** su trabajo **frecuentemente**
- Cada **integración** es **verificada** por un **build automático** que **detecta errores** de integración tan pronto como sea posible.

# Importancia de la Integración Continua

- Permite reducir riesgos
- Mejora la calidad del software
- Permite auto-detección de errores
- Frecuentemente se ejecutan las pruebas unitarias, funcionales, integración, ...
- Facilita el proceso de despliegue de las aplicaciones y servicios.

# Estructura de la Integración Continua



# Servidor de Integración Continua

- Constantemente monitorean los repositorios y apenas detectan un nuevo cambio, inician un nuevo ciclo de build.
- Involucra:
  - Compilación del código
  - Ejecución de las pruebas
  - Verificación de violaciones de estándares (métricas)
  - Notificación de errores ante fallas
  - Generación de documentación, reportes, etc.

# Meta de la Integración Continua

- “anyone should be able to bring in a virgin machine, check the sources out of the repository, issue a single command, and have a running system on their machine”



Martin Fowler

Fuente: <http://martinfowler.com/articles/continuousIntegration.html>

# Meta de la Integración Continua

- “Cualquier persona debe poder capaz de entregar en una nueva maquina, check las fuentes del repositorio, emitir un solo comando command, y poder ejecutar el sistema en otra maquina”



Martin Fowler

Fuente: <http://martinfowler.com/articles/continuousIntegration.html>

- “A *build* is simply an object that identifies the manufactured software product. The source code and build scripts are the raw materials; the manufacturing process begins when a build is started; and the resulting libraries, executables, and other supporting files are the result. ”

Fuente: Microsoft

<http://msdn.microsoft.com/en-us/library/ms364061.aspx>



- “A *build* es simplemente un objeto que identifica al producto. El código Fuente y los build script son los insumos; el proceso de creación empieza cuando un build es iniciado; y el resultado es una librería, ejecutable, y otros archivos de auxiliares que son el resultante.”

Fuente: Microsoft

<http://msdn.microsoft.com/en-us/library/ms364061.aspx>

Un “build” es el producto generado por la realización de las tareas de compilación, prueba, inspección e instalación. Realizar un “build” tiene como resultado situar el código todo junto en un ambiente donde funciona de una manera cohesiva. ”

- “A *build* es simplemente un objeto que identifica al producto. El código Fuente y los build script son los insumos; el proceso de creación empieza cuando un build es iniciado; y el resultado es una librería, ejecutable, y otros archivos de auxiliares que son el resultante.”

Fuente: Microsoft

<http://msdn.microsoft.com/en-us/library/ms364061.aspx>

# Mejores Prácticas



Cada “commit” debe construir la línea principal en un “Integration Server”



Mantener “build” rápidos



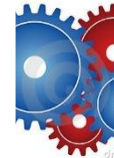
Probar en clones del entorno de producción



Hacer fácil el acceso a ejecutables



Todos observamos lo que sucede



Automatizar el despliegue



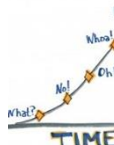
Mantener un “**repositorio único**” del producto



Automatizar las construcciones (“builds”)



Hacer “builds” auto-probados



Todos “remitimos” (commit) hacia línea principal cada día.

# ¿Por donde debemos iniciar?

Software para control de versiones

- svn, cvs, clearcase, git, etc.

Escriba unidades de prueba para el código

- Nunit, JUnit

Ponga en uso alguna herramienta de CI

- Jenkins

Utilice IC desde el inicio del proyecto, luego es complicado

Sea constante con la práctica de CI.

# ¿Qué hago para que CI sea efectiva?

Todo el proyecto  
mantenido en  
un solo  
repositorio.

Automatice las  
operaciones de  
build (las  
herramientas  
actuales poseen  
variantes)

Hacer commits  
(checkint) de los  
cambios cada  
día.

Asegure la  
consistencia de  
sus cambios con  
tests cases antes  
de hacer un  
commit.

Asegure que la  
creación de  
build sea rápida

Pruebe en un  
ambiente similar a  
producción

Asegure que el  
último build esté al  
alcance de los  
interesados

Todos deben tener  
visibilidad sobre lo  
que esté pasando  
(notifique: vía correo  
u otro medio – ver  
video)

Automatizar el  
“despliegue de  
aplicaciones”

# Ejemplos de Integración Continua



<https://sourceforge.net/projects/cruisecontrol/files/CruiseControl/2.8.4/>



<https://sourceforge.net/projects/ccnet/files/CruiseControl.NET%20Releases/>



Hudson (<http://hudson-ci.org/latest/hudson.war>)



Apache Continuum (<http://continuum.apache.org/download.html>)

# Otros Ejemplos de CI Server



CABIE



## Team Foundation Server



Rational Team Concert

A collaborative software development environment