

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y Sistemas
Análisis y Diseño de Sistemas 2
Ing. William Samuel Guevara Orellana
Aux. Samuel Rolando Matheu Macajola

Diego José Prera
201314280

HOJA DE TRABAJO 1: PARTE 1

¿QUÉ ES UN SISTEMA DE CONTROL DE VERSIONES?

Se puede definir como un sistema que se encarga de administrar los cambios que se realizan dentro de un producto de software.

¿QUÉ ES UN REPOSITORIO?

En su definición más general, un repositorio es un lugar donde cosas son almacenadas. Dependiendo del contexto, un repositorio podrá definirse de distintas formas. En el caso de los sistemas de control de versiones, un repositorio es un espacio virtual del almacenamiento donde va a ser alojado el proyecto de software junto con las distintas versiones de él, para luego ser accedidas cuando sean necesarios.

EN EL CONTEXTO DE SISTEMAS DE CONTROL DE VERSIONES, DEFINIR CON SUS PROPIAS PALABRAS LOS SIGUIENTES CONCEPTOS:

ÁRBOL

Un árbol es la estructura entera de los ciclos de cambios del proyecto. Cada nodo dentro de él es una versión, y este árbol puede dividirse en ramas que siguen su propia línea de versiones. A veces también se les llama árbol a las ramas.

REVISIÓN

Depende del sistema de control de versiones. En Git, una revisión es un "id" que se usa de parámetro para referenciar a un objeto en git, que usualmente es un commit. En SVN, una revisión es un nuevo estado el árbol del sistema luego del que el repositorio aceptara un commit. En otros casos, una revisión puede ser una versión mayor del sistema o hasta puede ser sinónimo de versión.

RELEASE

Es aquella versión del sistema que está listo para ser lanzado al público.

RAMA

Es la copia de un punto en el árbol que permite hacer un trabajo paralelo a la rama original. El fin de realizar ramas es la de poder realizar distintos cambios desde un punto sin afectar el trabajo original, para luego combinarlo con la rama original o tener un trabajo totalmente paralelo a este (cambios específicos a un cliente).

ETIQUETA

Es una marca en un punto en la historia de proyecto que se considere importante. Se utiliza más que todo para indicar versiones.

VERSIÓN

Es un punto en el árbol del sistema que consiste en un grupo de cambios o nuevas funcionalidades agregadas a una versión anterior.

COMMIT

También llamado “check in”, es el proceso de escribir los cambios realizados en la copia local al repositorio.

UPDATE

Es el proceso de combinar los cambios realizados en el repositorio hacia la copia local.

CHECK OUT

Es el proceso de creación de una copia local desde el repositorio.

MERGE

Es la operación de combinar dos partes con cambios en uno solo. Usado normalmente para combinar una rama con la rama principal o en el update.

DIFERENCIA ENTRE TRUNK Y BRANCH

Trunk es la línea principal de desarrollo en el proyecto. Es en donde se encuentra el código principal del proyecto, y tiende a ser siempre el que más estable se encuentre.

Branch o rama, es una desviación del trunk (o hasta de otra rama) en un punto en específico, donde se aplican algunos cambios que no quieren que se apliquen a la línea principal, pero que en un futuro puede ser combinado si es necesario.

Lo que normalmente se hace es que se mantiene el Trunk como el código principal del proyecto, y los branches en líneas paralelas en donde se aplican otros tipos de cambios (arreglo de bugs, experimentación, entre otros). Los cambios, luego de ser revisados correctamente, luego son integrados a la línea principal, y se continua con el proceso.

¿QUÉ ES Y PORQUE ES NECESARIA LA INTEGRACIÓN CONTINUA?

El proceso de integración continua es una práctica de desarrollo de software donde los miembros de un equipo de trabajo integran su trabajo de forma frecuente, muchas veces de forma diaria. Cada integración luego es verificada por un sistema de construcción y testeo, que verifica si existen errores e indicarlos de la forma más rápida posible.

Es necesario porque provee de distintos beneficios, además de lograr reducir distintos problemas que surgen en un proceso de desarrollo de software. Algunas ventajas que tiene este sistema son:

- Reduce el trabajo y el tiempo de las integraciones.
- Incrementa la visibilidad, lo que logra mejor comunicación.
- Reducción de riesgos.
- Iteraciones más rápidas.
- Mejor retroalimentación para toma de decisiones.
- Reducción de errores, además de aumentar la facilidad de encontrarlos.

DIAGRAMA DE PROCESO DE INTEGRACIÓN CONTINUA

