Unidad 4. Buenas prácticas para los modelos de prueba de software.

4.6 Llevar a cabo el control de cambios y versiones.

4.6 Llevar a cabo el control de cambios y versiones.

Una buena práctica en el desarrollo de software es llevar a cabo el control de versiones.

El control de versiones requiere de rastrear y gestionar los cambios en el código de software.

Para llevar a cabo el control de versiones actualmente se cuenta con herramientas o sistemas de control de versiones.

Las herramientas o sistemas de control de versiones son software que ayudan en la gestión de los cambios del código que el equipo de desarrollo va realizando.

Las herramientas de gestión de cambios apoyan de manera significativa en los desarrollos rápido al llevar a cabo el control de las versiones lo que permite que el equipo de desarrollo trabaje más rápido logrando implementaciones de manera exitosa.

Buenas prácticas de software

En el control de versiones se realiza un seguimiento de todas las modificaciones en el código.

Se pueden comparar las versiones del código, lo que ayuda a detectar y resolver errores.
Se minimizan las interrupciones durante el desarrollo.

Se debe tomar en cuenta que para todo proyecto de software el código fuente es muy por lo tanto se debe proteger.

Un componente, proceso o función de un proyecto se conforman de código y se organizan en carpetas con una estructura de árbol.

Buenas prácticas de software

El trabajo de varios desarrolladores no se ve afectado pues mientras unos trabajan en desarrollos de funciones nuevas otros solucionan errores, cada uno trabaja en las ramas de carpetas del árbol sin provocar conflictos.

El control de versiones realiza un seguimiento de todos los cambios individuales y del equipo.

Por lo que evita que el trabajo concurrente entre en conflicto.

Se recomienda hacer las pruebas cuando se concluya el desarrollo de módulos, funciones o procesos, hasta concluir la versión final.

Los cambios de versiones deben incluir el autor, la fecha y notas escritas sobre el propósito de cada cambio.

Con lo anterior se logra volver a las versiones anteriores ayudando a analizar la causa de los cambios en la raíz identificando la fuente de los errores.

El trabajo cotidiano de programación del software activa siempre una versión que se continua en la siguiente sesión de trabajo.

4.6.1 Herramientas para el control de cambios y versiones.

a) Sistemas de control de versiones locales.

Cuando se trabaja de manera individual generalmente el control de versiones y cambios se lleva a cabo haciendo una copia de la carpeta de desarrollo en el computador personal o local indicando la fecha de modificación para identificar la última versión guardada.

b) Sistemas de control de versiones centralizados.

Cuando se trabaja en proyectos grandes y con un equipo de desarrollo es necesario poder contar con un sistema colaborador de control de versiones centralizado.

4.6.1 Herramientas para el control de cambios y versiones.

c) Sistemas de control de versiones distribuidas.

En este sistema cada desarrollador contiene una copia completa del proyecto de forma local y en su conjunto cualquier repositorio permite compartir información.

4.6.1 Herramientas para el control de cambios y versiones.

Los sistemas de control de versiones (VCS) se conocen como herramientas de SCM (Gestión del código fuente) o RCS (Sistema de control de revisiones).

Una de las herramientas de VCS más utilizadas hoy en día es Git.

Git es un VCS distribuido de código abierto por lo tanto gratuito.

Git guarda un snapshot (instantánea) del estado de cada archivo.

Si el archivo no ha cambiado, no crea una nueva copia, sino que crea una referencia al archivo original.

4.6.1 Herramientas para el control de cambios y versiones.

Con Hit se logra:

Auditoría completa del código, identificando en todo momento quién acceso al código.

Permite visualizar el control de los cambios del proyecto.

Permite regresar a los desarrollos anteriores de forma rápida.

El control de versiones lo realiza por medio de etiquetas.

La seguridad es una característica fundamental y Hit la tiene ya que todas las estructuras internas de datos irán cifradas con el algoritmo SHA1.