

Administración de la configuración de Software

Cuarta sesión: Herramientas de SCM

Héctor José Fierros López

Mayo 2017



A cubrir hoy...

Propósito de las herramientas

El plan definirá las herramientas que se usarán.
Es posible llevar el proceso "a mano" siempre y cuando se defina un mecanismo de manejo de versiones. (Por ejemplo, incluyendo la versión en el nombre del archivo, usando banderas de cambio en el código, etc)
Sin embargo existen herramientas de software que facilitan esta tarea

Las funciones de el SW SCM

Una herramienta de SCM debe:

- ▶ Manejar la identificación de los AC y sus versiones.
- ▶ Facilitar la colaboración entre los desarrolladores.

Una herramienta de SCM puede verse como una BD.

Controladas localmente

Administración de la
configuración de
Software
Héctor José Fierros
López

Los artículos de configuración y los deltas estan almacenados de manera local en el sistema.

Los desarrolladores tienen que conectarse a este sistema, y trabajar ahí.

Ejemplos:

- ▶ RCS
- ▶ SCCS

Cliente/Servidor

Administración de la
configuración de
Software
Héctor José Fierros
López

Los artículos de configuración son almacenados en un servidor y los desarrolladores se conectan a través de un cliente. Los cambios son coordinados por el software servidor.

Ejemplos:

- ▶ CVS
- ▶ Subversion
- ▶ Accurev
- ▶ ClearCase
- ▶ Perforce
- ▶ VSS

Modelo distribuido

Administración de la
configuración de
Software
Héctor José Fierros
López

Cada desarrollador trabaja en su propio repositorio local, y los cambios y modificaciones son compartidos en pasos separados.

Ejemplos:

- ▶ Mercurial
- ▶ git
- ▶ arch
- ▶ Fossil
- ▶ VS Team Service
- ▶ TeamWare

git

Administración de la
configuración de
Software
Héctor José Fierros
López

Git fué desarrollado a fin de sustituir BitKeeper. Sus objetivos son:

- ▶ Velocidad
- ▶ Simplicidad
- ▶ Desarrollo no lineal
- ▶ Distribuido
- ▶ Manejo de proyectos grandes.

Usando git en un SCM

Administración de la
configuración de
Software

Héctor José Fierros
López

Preparando un ejemplo muy sencillo:

- ▶ Creamos un programa muy sencillo en C (hello world?)
- ▶ Crea un archivo de `readme.txt`
- ▶ Crea un subdirectorio, donde pondras otro archivo de texto mas, llamado `doc.txt`.

Estos tres archivos representan nuestros artículos de configuración.

Iniciando el baseline

Administración de la
configuración de
Software

Héctor José Fierros
López

Al instalar git, en windows se instalan tres comandos en el menú:

`git-bash` Es un shell tipo bash de unix, con varios comandos de unix disponibles.

`git-cmd` Es un shell tipo CMD de windows, donde el path incluye los ejecutables de git.

`git-gui` Un interface gráfico para ver el trabajo con git.

En linux y macOS, los comandos están disponibles a través del shell.

Creando el baseline de manera local

Administración de la
configuración de
Software

Héctor José Fierros
López

Para crear el baseline:

- ▶ En `git-bash` cambiate al directorio donde estan tus archivos.
- ▶ Escribe el comando `git init`. Esto crea un repositorio local vacio de git.
- ▶ Usa el comando `git status` entre comandos.
- ▶ Escribe el comando `git add .` Esto añade todos nuestros archivos al repositorio de git.
- ▶ Escribe el comando `git commit`. Esto promueve los archivos del baseline en el repositorio.
- ▶ El `git commit` lanza una ventana de vim, donde escribiras la razón del commit.

Entendiendo el repositorio

Administración de la
configuración de
Software

Héctor José Fierros
López

- ▶ El comando `git ls-tree -r HEAD` da un listado de los archivos.
- ▶ El comando `git log` da una historia del sistema.
- ▶ La meta-información sobre el repositorio se almacena en el directorio `.git`

Haciendo cambios: Nuevos archivos

Administración de la
configuración de
Software
Héctor José Fierros
López

- ▶ Crea un nuevo archivo (`test.txt`) en el arbol.
- ▶ Usa el comando `git status`. Muestra los cambios en el arbol.
- ▶ Usa el comando `git add test.txt`. Pasa la parte al control de git.
- ▶ Usa el comando `git status`. Muestra las partes listas para promover.
- ▶ Usa el comando `git commit`. Hace la promoción.

Haciendo cambios: Modificando archivos

Administración de la
configuración de
Software
Héctor José Fierros
López

- ▶ Modifica uno de los archivos en el arbol.
- ▶ Usa el comando `git status`: Muestra los cambios en el arbol.
- ▶ Usa el comando `git add -A`: Pasa los cambios al control de git.
- ▶ Usa el comando `git status`: Muestra las partes listas para promover.
- ▶ Usa el comando `git commit`: Hace la promoción.

Checando la historia

Administración de la
configuración de
Software
Héctor José Fierros
López

- ▶ El comando `git log` da un histórico de los cambios
- ▶ El comando `git show` da un vistazo de los cambios en cada commit
- ▶ El comando `git diff` saca las diferencias en los archivos antes de hacer commit.

Branches

Administración de la
configuración de
Software
Héctor José Fierros
López

- ▶ Cada commit cambia el estado del repositorio.
- ▶ Un branch es un apuntador a uno de los diferentes estados.
- ▶ Hay una rama principal llamada *master*
- ▶ El comando `git branch` sin parametros, lista las branches.
- ▶ El comando `git branch nuevarama`, crea un nuevo branch, llamado nuevarama.
- ▶ El comando `git checkout nuevarama` hace que el nuevo branch sea el que recibe los commits.
- ▶ El comando `git merge` Une diferentes branches.

Remotos

Administración de la
configuración de
Software

Héctor José Fierros
López

La parte mas poderosa del git es la capacidad de manejar datos distribuidos.

- ▶ Un repositorio "local" en un servidor de archivos.
- ▶ Un repositorio remoto, a través de:
 - ▶ ssh
 - ▶ ftp
 - ▶ http
 - ▶ git
- ▶ El comando `git clone` clona un repositorio remoto a un directorio.
- ▶ Los comandos `git pull` y `git push` "bajan" y "suben" los cambios al remoto.

Actividad de aprendizaje

Administración de la
configuración de
Software

Héctor José Fierros
López

- ▶ Usando git identifica...
 - ▶ ... como instalar
 - ▶ ... que comando usar para crear un baseline
 - ▶ ... que comando(s) usar para hacer una promoción
 - ▶ ... que comandos usar para hacer un branch
- ▶ repite usando mercurial, subversion y/o otro sistema de control de versión