

Gestión de la Configuración (SCM)



Introducción a la Ingeniería de Software

Temario

- Configuración del software
- Gestión de la Configuración
 - Versiones
- Control de Cambios
 - Línea base
- Auditoria de la configuración
- Herramientas
- Estrategias de Branch
- Antipatronos
- Tips

Problemas

- ¿Cambios hechos por distintos programadores se pierden?
- ¿Que versión hay que instalar en el cliente?
- Un error corregido, ¿reaparece?
- ¿Se implementó un cambio que no estaba confirmado?
- ¿Cuales son los fuentes que se corresponden con un ejecutable?
- ¿Cual es la ultima versión del Manual de Usuario?

Configuración

- Configuración del software: Los elementos que componen toda la información producida como parte del proceso de ingeniería de software
 - @ Fuentes
 - @ Ejecutables
 - @ Documentos
 - @ Datos, etc
- A medida que los elementos cambian se obtienen nuevas versiones, las cuales se deben identificar de forma única
- Suele ser necesario recuperar versiones antiguas

Gestión de la Configuración (SCM)

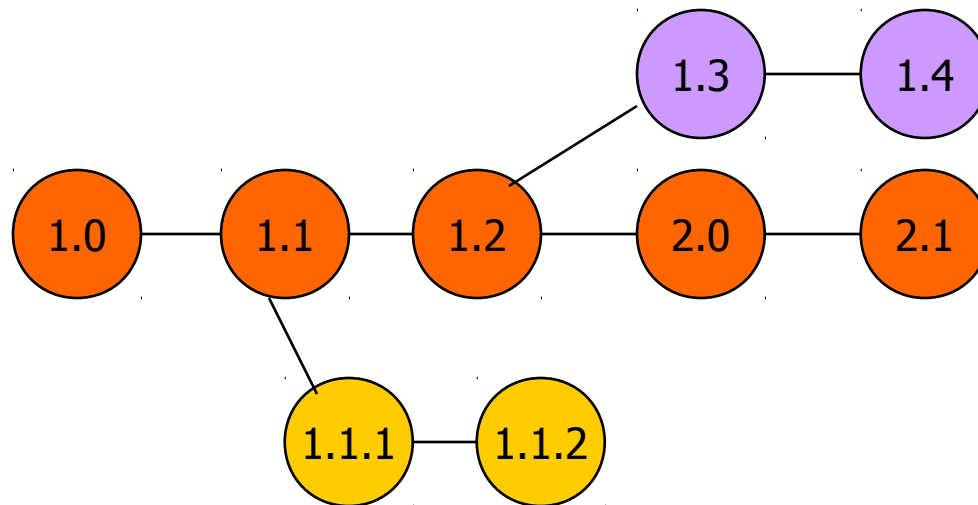
- Identifica, organiza y controla las modificaciones del software a través del ciclo de vida.
- Actividades de SCM:
 - Identificación de elementos
 - Control de versiones
 - Control de cambios
 - Auditar la configuración
 - Generación de informes
- Interesa identificar los elementos de configuración y localizarlos, seleccionando la versión apropiada, saber su historia y la razón de sus cambios
- Seguir la evolución del producto, administrar los requerimientos de cambio e implementarlos en forma consistente

Identificación de elementos

- Cada elemento se identifica de forma única
- Cada elemento consta de :
 - Nombre: Texto sin ambigüedad
 - Versión
 - Tipo: documento, programa, datos, etc
 - Proyecto
 - Información del cambio o la versión

Control de versiones

- Combina los procedimientos y herramientas para gestionar las versiones de los elementos
- Se puede versionar asociando un numero a cada versión



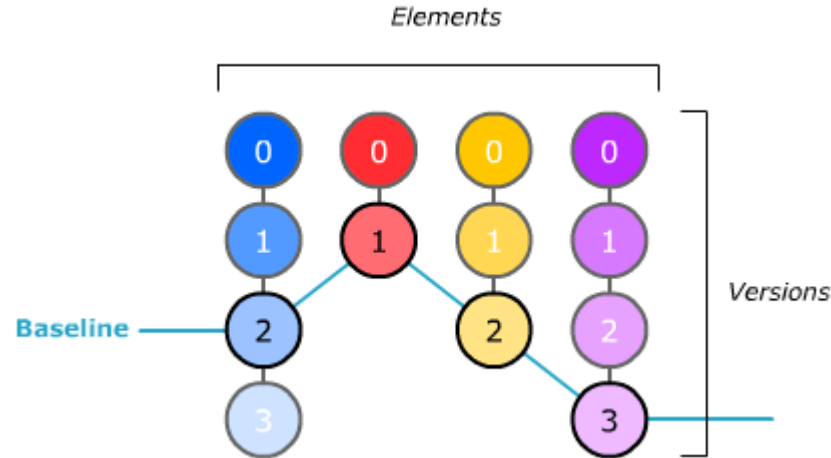
Controlar el Cambio

■ Línea Base

- Elemento de configuración que se ha revisado formalmente y que se ha llegado a un acuerdo
 - Sirve como base para desarrollos posteriores y puede cambiarse solo a través de los procedimientos de control de cambios
- Un elemento de configuración se convierte en línea base si fue revisado y aprobado
- Un cambio es el paso de una línea base a la siguiente

Controlar el Cambio

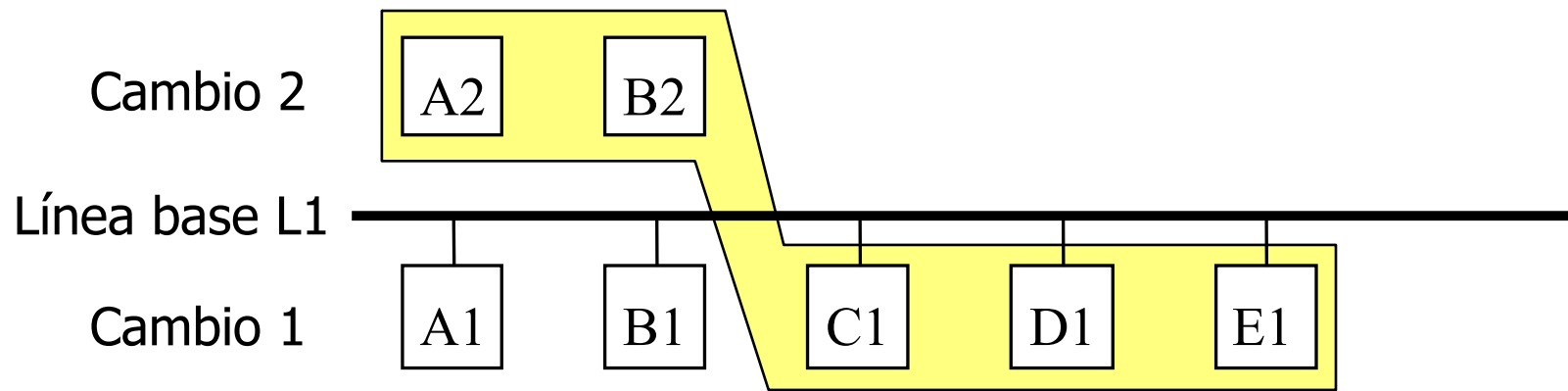
- Los elementos de la configuración se cambian rápida e informalmente hasta que en un momento se convierte en línea base, a partir de ahí el cambio se controla mediante procedimientos formales



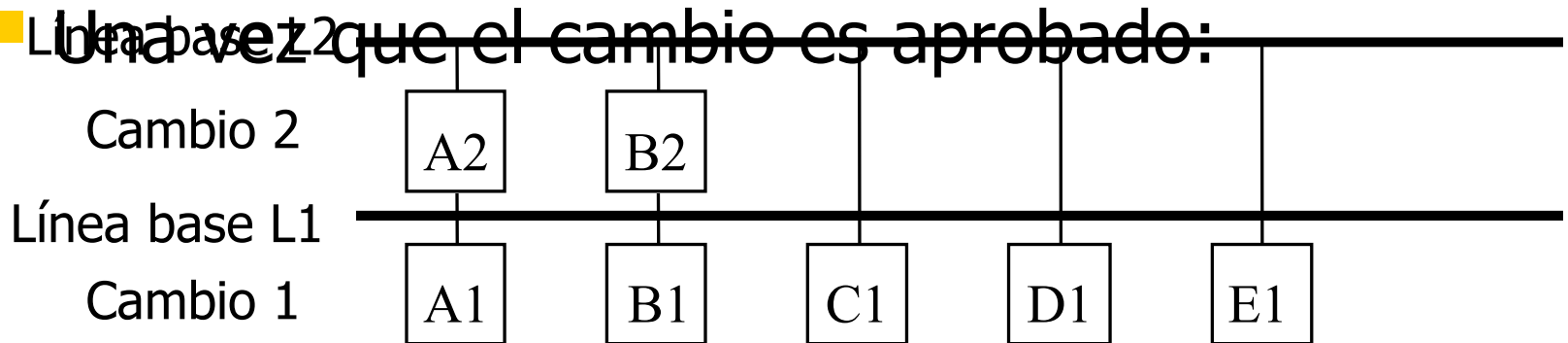
- Los desarrollos posteriores se hacen a partir de elementos en línea base

Línea base (baseline)

■ Controlar el cambio:



■ Una vez que el cambio es aprobado:



Las Líneas base permiten

- Ir atrás en el tiempo y reproducir el entorno de desarrollo en un momento dado del proyecto
- Trazabilidad: Permiten establecer relaciones de predecesor-sucesor entre artefactos. (seguir la pista y correspondencia)
 - Definición de Requerimientos
 - Especificación
 - Módulos de Diseño
 - Código que implementa los módulos
 - las pruebas para verificar la funcionalidad
 - los documentos que describen el sistema
- Comparar el contenido de una línea base contra otra

Registros

■ Formulario

■ Seguimiento de cambios

| Change Request Form | |
|--|----------------------------------|
| Project: Proteus/PCL-Tools | Number: 23/94 |
| Change requester: I. Sommerville | Date: 1/12/98 |
| Requested change: When a component is selected from the structure, display the name of the file where it is stored. | |
| Change analyser: G. Dean | Analysis date: 10/12/98 |
| Components affected: Display-Icon.Select, Display-Icon.Display | |
| Associated components: FileTable | |
| Change assessment: Relatively simple to implement as a file name table is available. Requires the design and implementation of a display field. No changes to associated components are required. | |
| Change priority: Low | |
| Change implementation: | |
| Estimated effort: 0.5 days | |
| Date to CCB: 15/12/98 | CCB decision date: 1/2/99 |
| CCB decision: Accept change. Change to be implemented in Release 2.1. | |
| Change implementor: | Date of change: |
| Date submitted to QA: | QA decision: |
| Date submitted to CM: | |
| Comments | |

■ Planilla

■ Historial de todos los cambios realizados

Controlar el Cambio

solicitar el cambio mediante el formulario de solicitud de cambio

analizar la solicitud de cambio

si el cambio es válido entonces

- se evalúa como será implementado el cambio

- se evalúa el costo del cambio

- se envía la solicitud al comité de control de cambios

- si el cambio es aceptado entonces

- repetir

- realizar cambios en el software

- enviar software con cambios para aprobación de calidad

- hasta que la calidad del software sea la adecuada

- crear una nueva versión del software

- de otro modo

- rechazar solicitud de cambio

de otro modo

- rechazar solicitud de cambio

Auditoria de Configuración

- Auditoria de la configuración: Verificar que en un momento dado, el Sistema en desarrollo es una colección de productos consistente y bien definida.
 - Determinar que todos los elementos de configuración están presentes en la línea base del Software, estableciendo la correctitud de la versión de cada elemento de configuración
 - Prevenir problemas

Generación de Informes

- Los informes intentan responder las siguientes preguntas:
 - @ ¿qué pasó?
 - @ ¿Quién lo hizo?
 - @ ¿cuándo?
 - @ ¿Qué mas se vió afectado?

Herramientas

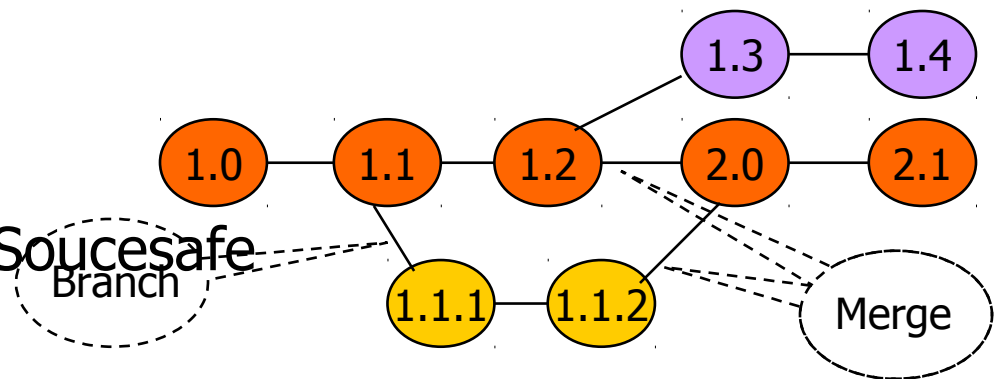
- Uso de repositorio – delta storage
- Modelo checkin/checkout
 - Modelo usado por algunas herramientas
 - Los archivos son almacenados y versionados en el repositorio
 - El usuario debe hacer un check out del archivo para ver o escribir
 - Los archivos modificados se devuelven al repositorio mediante checkin, creando una nueva versión
 - Tres formas de evolución:

@Versionado

@Merge

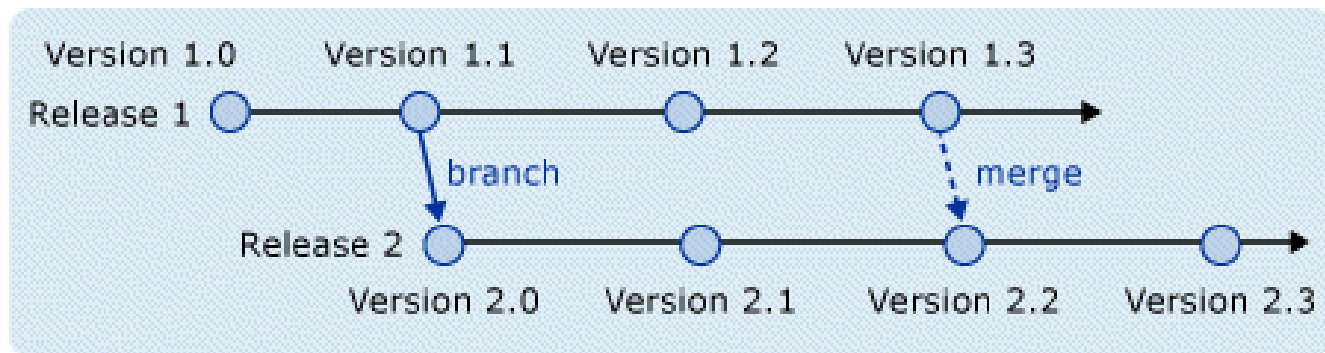
@Branch

- CVS, ClearCase, Visual SourceSafe



Estrategias de Branch

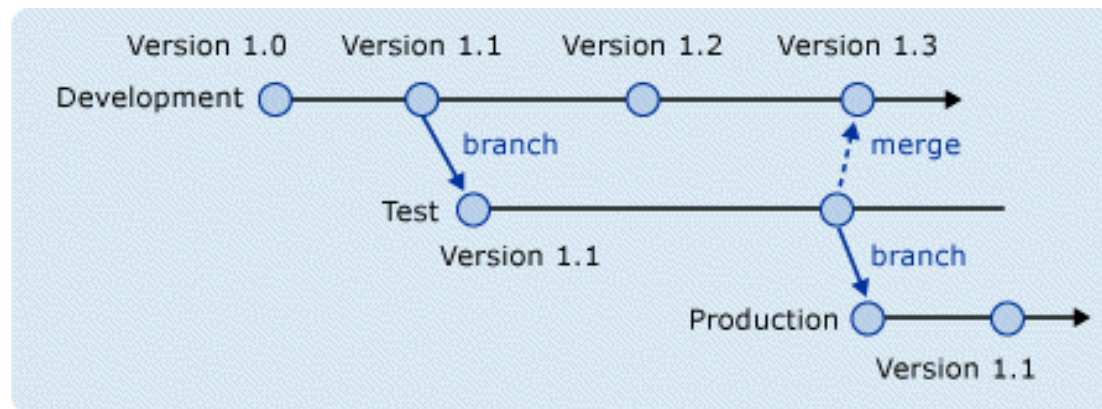
- Branch por release
 - Se ejecuta con cada nueva liberación.
 - Cada rama contiene toda la configuración.
 - Es posible realizar Merge entre versiones.



Estrategias de Branch

■ Code-Promotion Branch

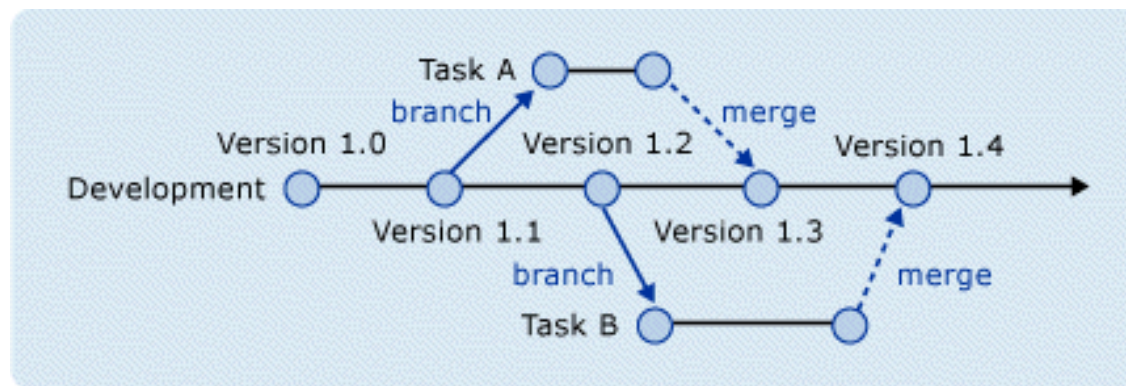
- Una versión de Desarrollo es ramificada hacia una de Testing, donde se ejecuta la integración y pruebas del Sistema.
- En caso de detectar un bug, se ejecuta un Merge con la versión de Desarrollo.
- Cuando termina el Testing, se genera un nuevo Branch, generando la configuración a ser entregada al cliente.



Estrategias de Branch

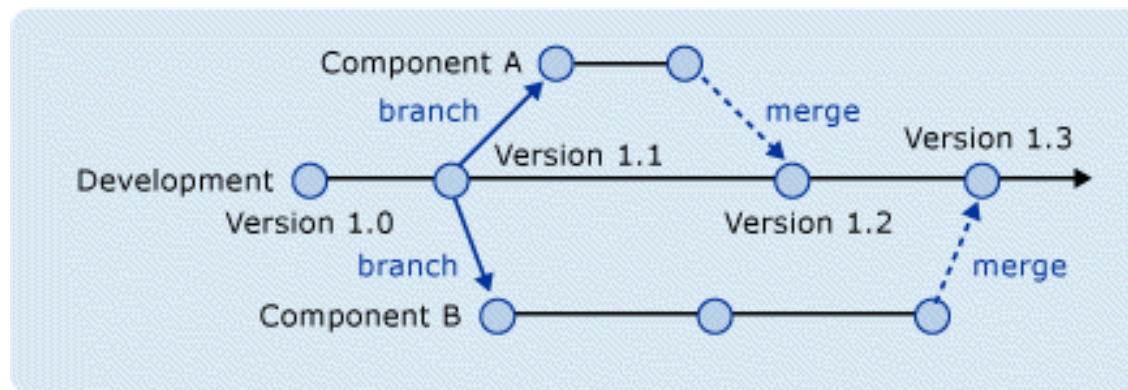
■ Branch por Tarea

- Para evitar la superposición de tareas y una pérdida de la productividad, se puede aislar una tarea en una rama separada.
- Debe ejecutarse un Merge una vez completada la tarea, no hacerlo puede repercutir en la productividad ganada al ejecutar el Branch.



Estrategias de Branch

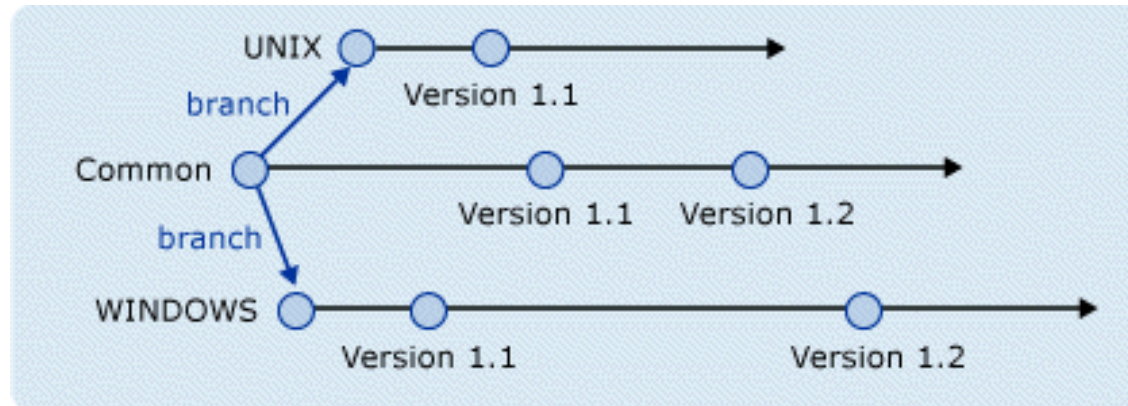
- Branch por Componente
 - Se alinean las ramificaciones con la arquitectura.
 - Cada rama corresponde a un componente (o subsistema)
 - A continuación, cada equipo fusiona sus códigos en la línea de desarrollo que sirve como la rama de integración.
 - Las interfaces deben estar bien definidas.



Estrategias de Branch

■ Branch por Tecnología

- Se alinean las ramas a las plataformas tecnológicas.
- El código común es administrado en una rama separada.
- Es probable que nunca se ejecutan fusiones.



Antipatronos

■ Merge-Paranoia

- Evitar la fusión a toda costa, por lo general a causa de un temor a las consecuencias.

■ Merge-Mania

- Gastar demasiado tiempo la fusión de configuración, en lugar de su desarrollo.

■ Big Bang Merge

- Aplazar la fusión hasta el final de las actividades de desarrollo y de intentar fusionar todas las ramas simultáneamente.

Antipatrones

■ Never-Ending Merge

- Fusiones continuas, porque siempre queda algo por integrar.

■ Wrong-Way Merge

- Fusionar una versión de componentes con una versión obsoleta.

■ Branch Mania

- Generar ramificaciones sin una versión aparente.

■ Cascading Branches

- Generar ramificaciones, pero nunca actualizar la línea base.

Antipatrones

■ Development Freeze

- Detener todas las actividades de desarrollo, mientras que ejecutan ramificaciones, fusiones, o la construcción de nueva línea base.

■ Berlin Wall

- Las ramificaciones dividen al equipo de desarrollo, en lugar de dividir su trabajo.

Tips

- SCM es la gestión de los cambios a productos de software
- Los documentos deben seguir una nomenclatura común.
- Se deben registrar la información de cambios y sus solicitudes.
- Un plan coherente de identificación de versiones debe ser elaborado.
- Las liberaciones del Sistema deben incluir código, datos, archivos de configuración y documentación.
- Existen herramientas para dar soporte a las actividades de SCM