Gestión de la configuración

¿Por qué necesitamos gestionar la configuración de un sistema? Problemas típicos:

- Se pierden los cambios hechos por otros programadores
- No sabemos qué versión debemos instalarle al cliente
- Reaparece un error ya corregido
- Se implementó un cambio que no estaba confirmado
- Se pierde el rastro de los fuentes que se corresponden con un ejecutable
- Se desconoce la versión del manual de usuario

Configuración

La configuración de un software son los elementos que componen toda la información producida por el proceso de ingeniería de software. Esto incluye: 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 anteriores, lo que muestra el por qué de la necesidad de gestionar los elementos.

Gestión de la configuración

Identifica, organiza y controla las modificaciones del software a través del ciclo de vida.
Sus actividades comprenden:

- Identificación de elementos
- Control de versiones
- Control de cambios
- Auditoría de la configuración
- Generación de informes

La gestión sirve para identificar los elementos de configuración y localizarlos, seleccionar la versión apropiada, saber su historia y la razón de sus cambios. Además, sigue la evolución del producto, administra los requerimientos de cambio y los implementa en forma consistente.

Identificación de elementos

Cada elemento se identifica de forma única y 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 número a cada versión.

Control de cambios

Se establece una “Línea base” para un elemento. Esto es que se ha revisado formalmente y se ha llegado a un acuerdo entre el equipo de desarrollo. Sirve como base para desarrollos posteriores y solo se puede cambiar a través de los procedimientos de control.
Solo los elementos que fueron revisados y aprobados pueden convertirse en línea base. Un **cambio** se considera el paso de una línea base **a la siguiente**.

Los elementos de configuración se cambian rápida e informalmente hasta que en un momento se convierte en línea base. A partir de ahí los cambios se controlan mediante procedimientos formales. Todos los desarrollos posteriores se hacen a partir de elementos en línea base.

Las líneas base permiten:

- Ir atrás en el tiempo y reproducir el entorno de desarrollo en un momento dado del proyecto
- Trazabilidad: Establecimiento de relaciones entre predecesores-sucesores de elementos. Abarca:
    - 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 con otra

Registros

Se tiene un formulario para el seguimiento y una planilla con la historia de todos los cambios realizados.

Auditoría de la configuración

Verifica que en un momento dado, el sistema en desarrollo es una colección de productos consistente y bien definida.

- Se determina 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.
- Previene problemas

Generación de informes

Buscan responder las preguntas:

1. ¿Qué pasó?
2. ¿Quién lo hizo?
3. ¿Cuándo?
4. ¿Qué más se vió afectado?

Herramientas para la gestión de la configuración

- Repositorios (delta storage)
- Modelo checkin/checkout
    - Utilizado por algunas herramientas
    - Los archivos se almacenan y versionan en el repositorio automáticamente
    - El usuario realiza un checkout del archivo para verlo o editarlo
    - Los archivos se devuelven al repositorio mediante checkin, creando una nueva versión
    - Hay tres formas de evolución de los archivos:
        - Versionado
        - Merge
        - Branch
- CVS, ClearCase, Visual SourceSafe

Estrategias de Branch

Branch por release

Cada nueva liberación es un branch en donde cada rama contiene toda la configuración. Es posible realizar un merge entre versiones distintas (branches distintos).

Code-promotion branch

Se realiza un branch por cada versión de desarrollo que pasa a testing donde se ejecuta la integración y pruebas del sistema. En caso de detectar un bug se ejecuta un merge con la versión en desarrollo que se siguió produciendo. Al terminar el testing, se ejecuta un **nuevo** branch que resulta en la versión entregada al cliente.

Branch por tarea

Se aisla una tarea en una rama separada para evitar la superposición de tareas y pérdidas en la productividad. Una vez completada la tarea, se ejecuta un merge con el desarrollo, ya que de lo contrario se podría producir una pérdida en la productividad ganada, por cuestiones de dependencia de tareas, por ejemplo.

Branch por componente

Las ramificaciones se corresponden con la arquitectura. Cada rama corresponde a un componente o subsistema. Al terminar cada rama, se fusionan los códigos en la línea de desarrollo que trabaja como rama de integración. Se requiere que las interfaces estén bien definidas de antemano.

Branch por tecnología

Las ramas se corresponden con plataformas tecnológicas y el código en común es administrado por una única rama separada. Probablemente nunca se ejecuten merges.

Antipatrones

- Merge Paranoia
  Evitar la fusión a toda costa, normalmente por temor a las consecuencias.
- Merge Mania
  Demasiado tiempo en la gestión de la fusión en vez de desarrollar.
- Big Bang Merge
  Aplazar la fusión hasta el final del desarrollo e intentar fusionar todas las ramas simultáneamente.
- Never ending Merge
  Fusiones contínuas porque siempre queda algo por integrar.
- Wrong-way Merge
  Fusionar componentes con una versión obsoleta
- Branch Mania
  Generar ramificaciones sin una versión aparente.
- Cascading Branches
  Generar ramificaciones sin actualizar la línea base
- Development Freeze
  Detener el desarrollo mientras se ejecutan ramificaciones, fusiones, o la construcción de una nueva línea base.
- Berlin Wall
  Las ramificaciones dividen al equipo de desarrollo, en lugar de dividir solamente el trabajo.

Tips para SCM

- Es la gestión de cambios a productos de software
- Los documentos deben seguir una nomenclatura común
- Se debe registrar la información de cambios y sus solicitudes
- Se debe elaborar un plan coherente de identificación de versiones
- Las liberaciones del sistema deben incluír el código, los datos, los archivos de configuración y la documentación.
- Existen herramientas para dar soporte a la SCM