



Universidad Nacional de Tucumán



Universidad Nacional de Tucumán
Facultad de Ciencias Exactas y Tecnología

Ingeniería de Software II

Gestión de Configuración del Software

Mg. Héctor A. Valdecantos

Software Configuration Management (SCM)

SCM es acerca de cómo controlamos la evolución de un proyecto de software. Disciplina de la ing. de software que comprende **herramientas** y técnicas (procesos y metodologías) que una compañía usa para administrar el cambio de sus activos de software. [Bellagio, David, and Tom Milligan, 2005.]

SCM es una disciplina de ingeniería formal que, como parte de la administración general de la configuración del sistema, proporciona los métodos y **herramientas** para identificar y controlar el software a lo largo de su desarrollo y uso. Las actividades de SCM incluyen la identificación y el establecimiento de líneas de base; la revisión, aprobación y control de cambios; el seguimiento y el informe de dichos cambios; las auditorías y revisiones del producto de software en evolución; y el control de la documentación de la interfaz y del proveedor del proyecto SCM. [IEEE-828 1998]

Software Configuration Management

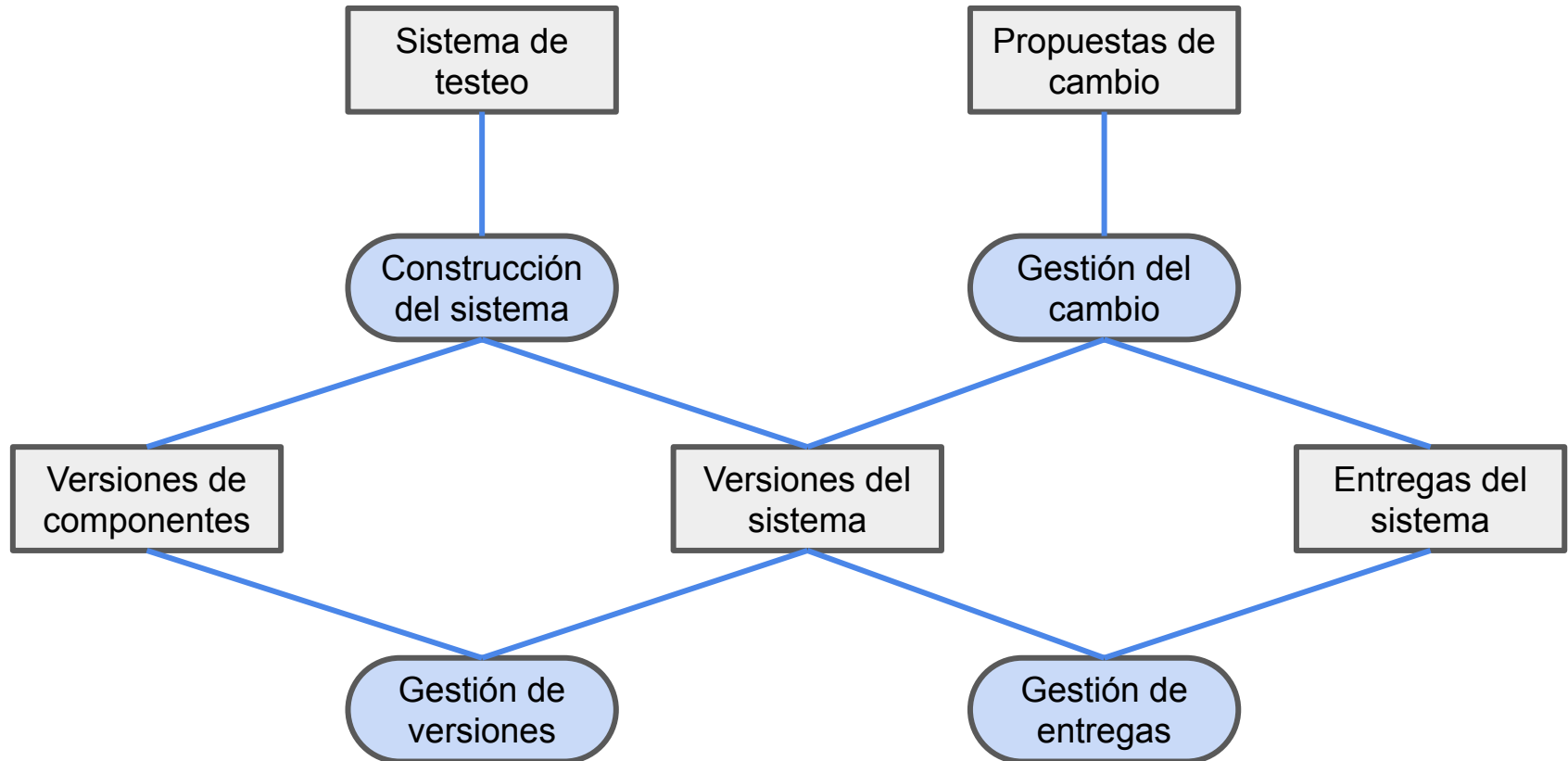
- SCM es útil para proyectos individuales, pero especialmente para grandes proyectos con un equipo numeroso
- Asegura el acceso a la información necesaria para el desarrollo del sistema de software y la no interferencia entre actividades desarrolladas en forma simultáneas
- Existen muchas herramientas que dan soporte a algunas de las actividades de CM, o a todo el proceso de gestión de configuración.
- Algunos autores consideran CM parte de la gestión de calidad
- CM es esencial para lograr algunas certificaciones de calidad:
 - ISO 9000, CMMI
 - La IEEE desarrolló estándares genéricos para CM

Actividades en SCM

Los sistemas de software siempre cambian, evolucionan, de otro modo, comienzan su fase de obsolescencia. CM incluye políticas, procesos, y herramientas para las siguientes actividades:

- **Gestión del cambio** (*change management*): Para mantener el rastro, analizar costo e impacto y decidir sobre los pedidos de cambios de los clientes y desarrolladores.
- **Gestión de versiones** (*version management*): Seguimiento de las múltiples versiones de sistemas o componentes y mecanismos para favorecer el trabajo concurrente.
- **Construcción del sistema** (*system building*): el proceso de construcción de los sistemas con sus los datos, librerías para su posterior integración o compilación.
- **Gestión de entregas** (*release management*): la preparación del producto o servicio de software para su puesta en producción

Actividades en SCM



Gestión del cambio (*change management*)

- Se completa un formulario de pedido de cambio (CRF)
 - Reporte de bug, nueva funcionalidad, mejora, etc
- El CRF debe ser aprobado y validado
 - bug duplicado; no es un bug, es una feature; funcionalidad ya implementada pero desconocida; etc
- Un CRF válido es evaluado: estimación de riesgo, estimación de costo, impacto de posibles soluciones (otros componentes)
- Mientras se procesa el CRF, se le va agregando información
 - fecha de apertura, aprobación; análisis de riesgo; severidad, costo; etc
- Una comisión de control de cambio, responsable de cómo evoluciona el producto (visión estratégica y organizacional). Los encargados del desarrollo del producto.

Gestión del cambio (*change management*)

- El software está sujeto a continuos pedidos de cambios (*change requests*) a medida que se descubren problemas, surgen funcionalidades nuevas, etc
- La gestión de cambios trata de la gestión de estos cambios y de asegurar que los cambios se implementen de una manera costo-efectiva
- La gestión del cambio debe incluir procesos para decidir qué problemas resolver, quién debe resolverlos y cuándo deben ser resueltos

Gestión de versiones (*version management*)

- Se establece un esquema de identificación de versiones
 - codelines y baselines
- Se favorece el trabajo concurrente de los desarrolladores
- Cualquier ítem de configuración de software es susceptible de ser versionado: doc. de diseño, código, test cases, etc.
- Es impracticable la gestión de versiones manual
 - Version Control System (VCS): identifican, guardan, y controlan el acceso de diferentes versiones
 - Identificación de entrega y versión
 - Gestión del almacenamiento
 - Registro del historial de cambios.
 - Desarrollo independiente de múltiples ítem de configuración
 - Soporte para múltiple proyectos

Gestión de versiones (*version management*)

Codeline (A)



Codeline (B)



Codeline (C)



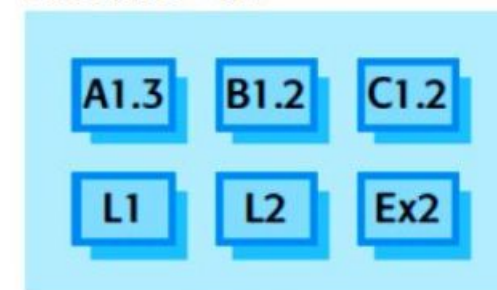
Libraries and External Components



Baseline - V1



Baseline - V2



Mainline

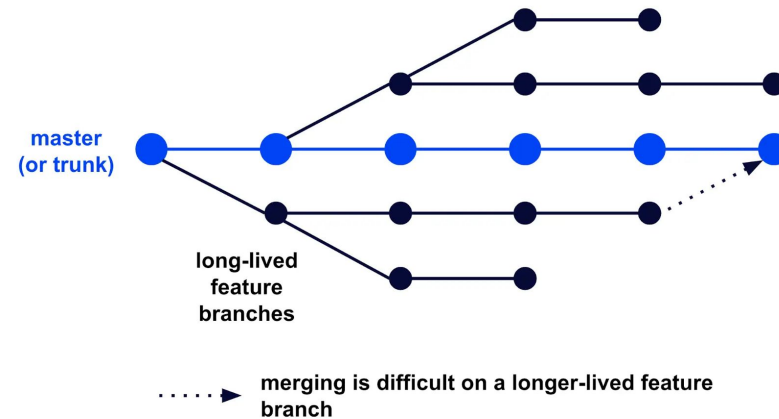
Construcción del sistema (*system building*)

Proceso para crear un sistema completo ejecutable por medio de la compilación y linkeo de componentes de sistemas, librerías externas, archivos de configuración, etc.

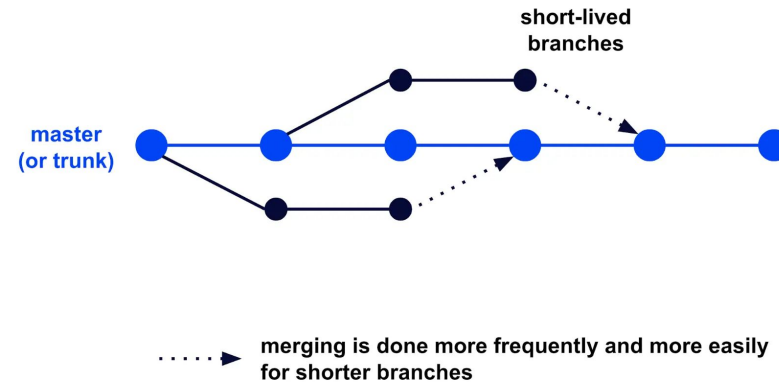
- Se trata de que este proceso sea lo más automático posible
 - El enfoque de integración continua
- El sistema de construcción del sistema debe comunicarse con el sistema de gestión de versiones para la construcción
- En lo general, hay tres ambientes posibles:
 - De desarrollo
 - De construcción
 - De producción

Construcción del sistema (*system building*)

Feature-branched development



Trunk-based development



Gestión de entregas (*release management*)

Una entrega (*release*) del sistema de software es una versión que es distribuida a los clientes.

- Típicamente se distinguen dos tipos de *releases*
 - Major release: con cambios significativos (ej: new feature)
 - Minor release: con cambios menores (ej: bug fixes)
- Cuando se produce un release se debe documentar de manera precisa su versión para poder recrear esa versión en un futuro
 - Versiones del código fuente, archivos de configuración, librerías, etc (todos los ítems de configuración usados)
- Para productos masivos, preparar un *release* puede ser costoso, incluye:
 - crear la distribución, la publicidad, la comercialización
 - Un release de un sistema no es sólo el código ejecutable

Gestión de entregas (*release management*)

Factor	Description
Technical quality of the system	If serious system faults are reported which affect the way in which many customers use the system, it may be necessary to issue a fault repair release. Minor system faults may be repaired by issuing patches (usually distributed over the Internet) that can be applied to the current release of the system.
Platform changes	You may have to create a new release of a software application when a new version of the operating system platform is released.
Lehman's fifth law (see Chapter 9)	This 'law' suggests that if you add a lot of new functionality to a system; you will also introduce bugs that will limit the amount of functionality that may be included in the next release. Therefore, a system release with significant new functionality may have to be followed by a release that focuses on repairing problems and improving performance.
Competition	For mass-market software, a new system release may be necessary because a competing product has introduced new features and market share may be lost if these are not provided to existing customers.
Marketing requirements	The marketing department of an organization may have made a commitment for releases to be available at a particular date.
Customer change proposals	For custom systems, customers may have made and paid for a specific set of system change proposals, and they expect a system release as soon as these have been implemented.

Bibliografía

~~Bellagio, David, and Tom Milligan. Software configuration management strategies and ibm® rational® clearcase®: a practical introduction. (Chapter 1), IBM Press, 2005.~~

Pressman, Roger S. Software engineering: a practitioner's approach. 7th edition. Palgrave Macmillan, (2010).

Ian Sommerville. Software Engineering (9th ed.). Addison-Wesley Publishing Company, USA, (2010).