

CAPÍTULO 7

GESTIÓN DE CONFIGURACIÓN DEL SOFTWARE

ACRÓNIMOS

| | |
|----------|-------------------------------------------------------------------------------|
| CCB | Tarjeta de Control de la Configuración |
| CM | Gestión de configuración |
| FCA | Auditoría de la Configuración Funcional |
| MTBF | Tiempo significativo entre fallos. |
| PCA | Auditoría de la Configuración Física |
| SCCB | Consejo de Control de Configuración del Software |
| SCI | Elemento de configuración de software |
| SCM | Gestión de la configuración del software |
| SCMP | Plan de gestión de la configuración del software |
| SCR | Petición de cambios del software |
| SCSA | Contabilidad del Estado de la Configuración del Software |
| SEI/CMMI | Instituto de Ingenieros Software/ Modelo de la Capacidad de Madurez Integrado |
| SQA | Garantía de calidad del software. |
| SRS | Especificación de Requisitos Software |
| USNRC | Comisión Reguladora de Energía Nuclear de los Estados Unidos |

INTRODUCCIÓN:

Un sistema se puede definir como una colección de componentes que se organizan con el objetivo de proporcionar una función o conjunto de funciones determinadas (IEEE 610.12-90). La *configuración* de un sistema son las características funcionales y/o físicas del hardware, firmware, software o una combinación de las mismas, según lo dispuesto en la documentación técnica y el resultado obtenido en un producto. (Buc96) También se puede considerar como una colección de versiones específicas de elementos de hardware, firmware o software que se combinan de acuerdo con un proceso de construcción específico para un satisfacer un propósito particular. Por tanto la *gestión de configuración* es la disciplina de identificar la configuración de un sistema en momentos diferentes con el propósito de controlar de una manera sistemática los cambios en la configuración y mantener la integridad y el seguimiento de de los cambios en la configuración durante el ciclo de vida del sistema. (Ber97) Se define formalmente (IEEE610.12-90) como “Una disciplina que establece dirección y seguimiento técnicos y administrativos a: la identificación y documentación de las características funcionales y físicas de un elemento de configuración, toma notas y produce informes de cambios en el proceso y en el estado de implementación y verifica el cumplimiento de los requerimientos especificados.”

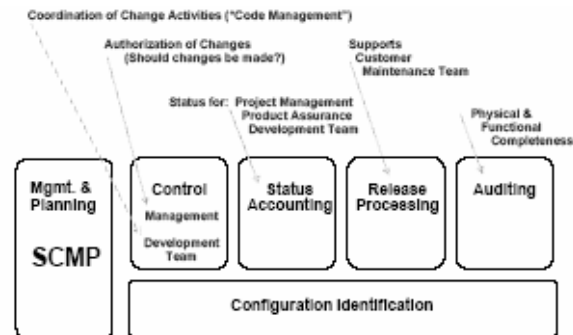
La *gestión de la configuración del software* (SCM) es un proceso que soporta el ciclo de vida del software (IEEE12207.0-96) que beneficia a la gestión de proyectos, las actividades de desarrollo y mantenimiento, las actividades de garantía y a los clientes y usuarios del producto final.

El concepto de gestión de configuración es aplicable a todos los elementos que se pueden controlar, aunque

existen algunas diferencias de implementación entre CM del hardware y CM del software.

La SCM esta íntimamente relacionada con la actividad de garantía de calidad del software (SQA). Tal y como se define en el AC de la Calidad del Software, los procesos de la SQA proporcionan la garantía de que los procesos y productos de software en el ciclo de vida del proyecto cumplen los requerimientos especificados gracias a la planificación, la promulgación y ejecución de un conjunto de actividades que proporcionan la confianza necesaria de que se está teniendo en cuenta la calidad al construir el software. Las actividades de la SCM ayudan a conseguir estos objetivos de la SQA. En el contexto de algunos proyectos (véase, por ejemplo, IEEE730-02), requerimientos específicos de las SQA requieren ciertas actividades de la SCM.

Las actividades de la SCM son: gestión y planificación de los procesos de la SCM, identificación de la configuración del software, control de la configuración del software, responsabilidad del estado de la configuración del software, auditoría de la configuración del software y gestión del lanzamiento y entrega del software.



La figura 1 muestra una representación estilizada de estas actividades.

Figura 1 Actividades SCM

El KA de la Gestión de Configuración del Software se relaciona con el resto de KAs, ya que el objetivo de la configuración del software es el producto construido y usado durante todo el proceso de ingeniería del software.

DIVISIÓN DE PUNTOS PARA LA SCM

1. Gestión del proceso de la SCM

La SCM controla la evolución e integridad de un producto identificando sus elementos, gestionando y controlando los cambios y verificando, guardando y produciendo informes de la información de configuración. Desde la perspectiva del ingeniero de software, la SCM facilita las actividades del desarrollo e implementación de cambios. El éxito de una implementación de la SCM requiere una planificación y gestión cuidadosas. Lo que al mismo tiempo requiere

que se conozca el contexto de organización y las restricciones impuestas en el diseño e implementación del proceso de la SCM.

1.1 Contexto de Organización para la SCM [Ber92:c4; Dar90:c2; IEEE828-98:c4s2.1]

Para planificar un proceso de la SCM para un proyecto se necesita comprender el contexto de organización y la relación entre los distintos elementos de la organización. La SCM interacciona con otras actividades o elementos de la organización.

Los elementos de la organización responsables de los procesos de soporte de la ingeniería del software se pueden estructurar de diferentes formas. Aunque la responsabilidad de realizar algunas de las tareas de la SCM se podría asignar a otra de las partes de la organización como por ejemplo la organización de desarrollo, normalmente es un elemento definido de la organización o un individuo especialmente designado quien tiene la responsabilidad general de la SCM.

El software se desarrolla frecuentemente como una parte de un sistema mayor que contiene elementos de hardware y firmware. En ese caso, las actividades de la SCM suceden en paralelo con las actividades de CM del hardware y firmware y debe ser consistente con la CM del sistema. Buckley [Buc96:c2] describe la SCM en este contexto. Tenga en cuenta que el firmware contiene hardware y software, así que son aplicables los conceptos de CM de ambos, hardware y software.

La SCM puede interactuar con la actividad de la garantía de la calidad de la organización en lo que se refiere a temas como la gestión de registros y de elementos no válidos. Respecto a gestión de registros, algunos elementos bajo el control de la SCM podrían ser también registros del proyecto, dependiendo del programa de garantía de calidad de la organización. Normalmente la gestión de elementos no válidos es responsabilidad de la actividad de garantía de la calidad; sin embargo, la SCM puede ayudar mediante el seguimiento e informes de

Figura 2 División de los puntos de función para el KA de la elementos de configuración del software que caigan en esta categoría.

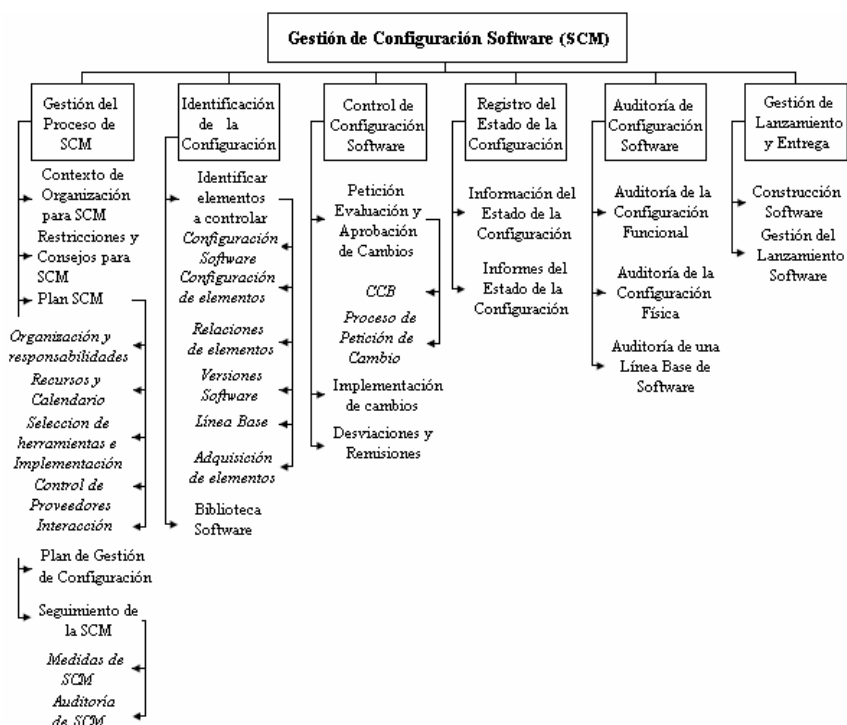
Quizás su relación más cercana sea con las organizaciones de desarrollo de software y mantenimiento.

Muchas de las tareas de control de configuración del software se realizan en este contexto. Frecuentemente las mismas herramientas proporcionan soporte para el desarrollo, mantenimiento y la SCM.

1.2 Restricciones y Consejos para el proceso de la SCM [Ber92:c5; IEEE828-98:c4s1,c4s2.3; Moo98]

Las restricciones y consejos para el proceso de la SCM pueden venir de diferentes fuentes. Las normas y procedimientos definidos a nivel corporativo o de la organización pueden tener influencia o prescribir el diseño e implementación de los procesos de la SCM en un determinado proyecto. Además, el contrato entre el proveedor y el cliente podría contener estipulaciones que afecten los procesos de la SCM. Por ejemplo, podría ser necesaria una auditoría de configuración o podría ser necesario poner ciertos elementos bajo el control de la CM. Cuerpos de regulación externos podrían imponer restricciones a productos de software que se vayan a desarrollar cuando estos puedan afectar potencialmente a la seguridad pública (véase, por ejemplo, USNRC1.169-97). Finalmente, el proceso del ciclo de vida del software elegido para un proyecto de software en particular y las herramientas elegidas para la implementación de dicho software, afectan el diseño e implementación de los procesos de la SCM. [Ber92]

Las “mejores prácticas”, como se reflejan en los estándares de la ingeniería del software publicados por varias organizaciones de estándares, se pueden usar como consejo para el diseño e implementación de un proceso de la SCM. Moore [Moo98] proporciona una guía para dichas organizaciones y sus estándares. Las mejores prácticas se reflejan también en modelos de



mejora del
Gestión de la Configuración del Software
proceso y valoración de procesos como el Modelo de la
Capacidad de Madurez Integrado del Instituto de la
Ingeniería del Software (SEI/CMMI) (SEI01) y el
estándar ISO/IEC15504 de la Valoración del Proceso de
la Ingeniería del Software (ISO/IEC15504-98).

1.3 Planificar la SCM

[Dar90:c2; IEEE12207.0-96 :c6.s2.1;
Som01:c29]

La planificación de un proceso de la SCM para un
proyecto dado debería ser consistente con el contexto de
la organización, las restricciones que sean aplicables, los
consejos comúnmente aceptados y la naturaleza del
proyecto (por ejemplo, tamaño lo crítico que sea). La
actividades más importantes cubiertas son:
Identificación del Configuración del Software, Control
de la Configuración del Software, Responsabilidad del
Estado de la Configuración del Software, Auditoría de la
Configuración del Software y la Gestión de
Lanzamiento y Entrega del Software. Además, se suelen
considerar puntos como organización y
responsabilidades, recursos y calendarios, selección de
herramientas e implementación, control de proveedores
y subcontratas y control de la interacción. Los resultados
de la planificación de actividades se registran en un Plan
de SCM, que normalmente está sujeto a revisión y
auditoría de la SQA.

1.3.1 Organización y responsabilidades de la SCM [Ber92:c7; Buc96:c3; IEEE828-98:c4s2]

Para prevenir confusión acerca de quien debe realizar
tareas de la SCM determinadas, se deben identificar
claramente las organizaciones involucradas en el
proceso de la SCM. Responsabilidades específicas para
una actividad o tarea de la SCM también deben ser
asignadas a organizaciones, bien por título o por
elemento de la organización. Se debe identificar también
la autoridad general y las vías de información de la
SCM, aunque se podría realizar como parte de la fase de
planificación de la garantía de la calidad o al nivel de la
gestión de proyectos.

1.3.2 Recursos y planificación de la SCM [Ber92:c7; Buc96:c3; IEEE828-98:c4s4; c4s5]

Planificar para la SCM ayuda identifica las necesidades
de personal y herramientas que se requieren para realizar
las tareas y actividades de la SCM. Aborda cuestiones
de planificación estableciendo las secuencias de tareas
de la SCM necesarias e identifica sus relaciones con los
planes de proyecto y los hitos establecidos en la fase de
planificación del proyecto. También se especifican las
necesidades de formación requeridas para la
implementación de los planes y formación de personal.

1.3.3 Selección e implementación de herramientas [Ber92:c15; Con98:c6; Pre01:c31]

Las actividades de la SCM son soportadas por diferentes
tipos de habilidad de herramientas y procedimientos
para su uso. Dependiendo de la situación, se pueden
combinar estas habilidades de herramientas con
herramientas manuales, herramientas automáticas que
proporcionan una habilidad simple de la SCM,
herramientas automáticas que integran una colección de
habilidades de la SCM (e incluso otras habilidades). o
entornos de herramientas integrados que cubren las
necesidades de múltiples participantes en el proceso de

ingeniería del software (por ejemplo, SCM, desarrollo,
V&V). El apoyo de herramientas automáticas se
comienza a ser más importante y difícil de establecer,
según los proyectos crecen en tamaño y el entorno del
proyecto se hace más complejo. Las habilidades de estas
herramientas proporcionan apoyo para:

- ♦ La biblioteca de la SCM
- ♦ Procedimientos de aprobación y de petición de
cambios del software (SCR)
- ♦ Tareas de gestión de cambios y código (y los
productos relacionados)
- ♦ Informes del estado de configuración del software
y reunión de medidas de la SCM
- ♦ Auditoría de la configuración del software
- ♦ Gestión y seguimiento de la documentación del
software
- ♦ Construcción del software
- ♦ Gestión y seguimiento de los lanzamientos del
software y su distribución

Las herramientas utilizadas en estas áreas, también
pueden proporcionar medidas para mejorar el proceso.
Royce [Roy98] describe siete medidas centrales útiles
para gestionar procesos de la ingeniería del software. La
información disponible de las varias herramientas de la
SCM es afín al indicador de Trabajo y Progreso de
Royce y a sus indicadores de calidad de Cambio de
Tráfico y Estabilidad, Ruptura y Modularidad,
Repetición del Trabajo y Adaptabilidad y TMEF
(Tiempo medio entre fallos) y Madurez. Los informes
para estos indicadores se pueden organizar de varias
maneras, como por elemento de configuración del
software o por tipo del cambio requerido.

La figura 3 muestra una asignación representativa entre
las habilidades de las herramientas y procedimiento a
Actividades de la SCM.

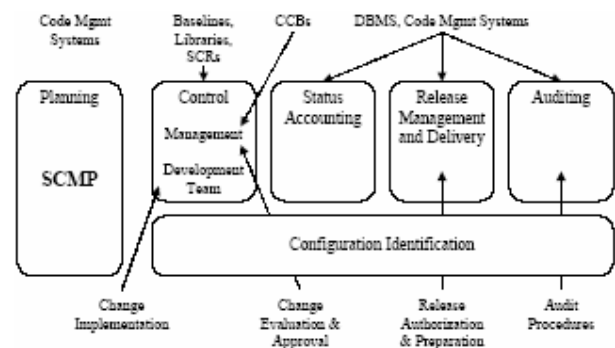


Figura 3 Caracterización de herramientas de SCM y
procedimientos relacionados

En este ejemplo, el sistema de gestión de código soporta
la utilización de bibliotecas de software al controlar el
acceso a los elementos de la biblioteca, coordinar las
actividades de múltiples usuarios y ayudar a hacer
cumplir los procedimientos operativos. Otras
herramientas soportan el proceso de construir software y
producir documentación de los elementos de software
contenidos en las bibliotecas. Herramientas para la
gestión de peticiones de cambios del software soportan
los procedimientos de control de cambios que se aplican
a elementos de software. Otras herramientas pueden

proporcionar gestión de bases de datos y habilidad para generar informes usados para las actividades de gestión, desarrollo y garantía de calidad. Como se ha mencionado antes, las habilidades de diferentes tipos de herramientas se podrían integrar en sistemas de la SCM, los cuales, a su vez, tienen un acoplamiento alto con otras actividades del software.

Al planificar, el ingeniero de software elige las herramientas de la SCM apropiadas para el trabajo. La planificación debe considerar los problemas que podrían surgir durante la implementación de dichas herramientas, particularmente si se necesita algún cambio cultural. Una introducción de los sistemas de la SCM y de consideraciones en su selección se puede encontrar en [Dar90:c3, AppA] se puede encontrar un caso de estudio de selección de un sistema de la SCM en [Mid97]. Información complementaria acerca de herramientas de la SCM se pueden encontrar en el AC de Métodos y Herramientas de la Ingeniería del Software.

1.3.4 Control Proveedores/Subcontratas [Ber92:c13; Buc96:c11; IEEE828-98:c4s3.6]

Un proyecto de software podría hacer uso o adquirir productos software que se hayan comprado, como compiladores u otras herramientas. La planificación de la SCM considera si y como se pondrán estos elementos bajo control de configuración (por ejemplo, integrados en bibliotecas de proyecto) y como se evaluarán y gestionarán los cambios y actualizaciones.

Consideraciones similares son aplicables al software subcontratado. En este caso, también se deben establecer los requerimientos de la SCM a ser impuestos a los procesos de la SCM del subcontratista como parte del subcontrato y los medios para monitorizar que se cumple con ellos. El último incluye consideraciones acerca de que información de la SCM ha de estar disponible para poder monitorizar, de una manera eficiente, que se cumplen los requerimientos.

1.3.5 Control de Interacción [IEEE828-98:c4s3.5]

Cuando un elemento de software interacciona con otro elemento de software o hardware, un cambio a cualquiera de los dos elementos puede afectar al otro. La planificación para los procesos de la SCM considera como se identificarán los elementos que interaccionan y como se gestionarán y comunicarán los cambios a dichos elementos. El papel de la SCM puede ser parte de proceso de nivel de sistema más general para el control y especificación de las interacciones y podría afectar a las especificaciones de las interacciones, planes de control de las interacciones e documentos de control de las interacciones. En este caso, la planificación de la SCM para el control de las interacciones ocurre dentro del contexto del proceso de nivel de sistema. Se puede encontrar una discusión del rendimiento de las actividades del control de las interacciones en [Ber92:c12].

1.4 Plan de la SCM [Ber92:c7; Buc96:c3; Pau93:L2-81]

Los resultados de la planificación de la SCM para un proyecto dado se reflejan en un Plan de Gestión de la Configuración del Software, que es un “documento vivo” que sirve como referencia para los procesos de la SCM. El documento se mantiene (o sea, se actualiza y aprueba) según vaya siendo necesario durante el ciclo de vida del software. Al implementar un SCMP,

normalmente es necesario desarrollar un conjunto de procedimientos subordinados más detallados, que definirán la forma en que se realizan requerimientos específicos durante las actividades del día a día.

Se pueden utilizar varias fuentes de información para encontrar consejo, basado en la información producida durante la actividad de planificación, acerca de la creación y mantenimiento de un SCMP, como en [IEEE828-98:c4]. Esta referencia proporciona requerimientos para la información que un SCMP ha de contener. También define y describe seis categorías de información de la GCS que se han de incluir en un SCMP:

- ♦ Introducción (propósito, extensión, términos usados)
- ♦ Gestión de la SCM (organización, responsabilidades, autoridades, normas aplicables, directivas y procedimientos)
- ♦ Actividades de la SCM (identificación de la configuración, control de la configuración, etc)
- ♦ Planificación de la SCM (coordinación con otras actividades del proyecto)
- ♦ Recursos de la SCM (herramientas, recursos físicos y recursos humanos)
- ♦ Mantenimiento del SCMP

1.5 Seguimiento de la Gestión del la Configuración del Software [Pau93:L2-87]

Después de que el proceso de la SCM de ha implementado, puede ser necesario un cierto nivel de seguimiento para asegurarse de que las previsiones de la SCM se llevan a cabo adecuadamente (véase, por ejemplo [Buc96]). Probablemente habrá requerimientos específicos de la SQA para asegurarse de que se cumplen los procesos y procedimientos específicos de la SCM. Esto podría requerir que una autoridad de la SCM se asegure de que aquellos que tengan responsabilidades asignadas realizan las tareas de la SCM definidas de una manera correcta. Este seguimiento podría ser realizado, como parte de una actividad de cumplimiento de auditoría, por la autoridad de la garantía de la calidad del software.

El uso de herramientas integradas de la SCM con posibilidades de control de procesos puede hacer la tarea de seguimiento más fácil. Algunas herramientas facilitan comprobar que el proceso se cumple al mismo tiempo que proporcionan al ingeniero de software la flexibilidad de adaptar procedimientos. Otras herramientas se aseguran de que el proceso se siga, dejando menos flexibilidad al ingeniero de software. Los requerimientos de seguimiento y los niveles de flexibilidad que se proporcionarán al ingeniero de software son criterios importantes durante la selección de herramientas.

1.5.1 Medidas y mediciones de la SCM [Buc96:c3; Roy98]

Se pueden asignar medidas de la SCM para proporcionar información específica acerca de la evolución del producto o una visión interna de como funcionan los procesos de la SCM. Un objetivo relacionado con el seguimiento de los procesos de la SCM es descubrir oportunidades para mejorar los procesos. Las mediciones de procesos de la SCM proporcionan un buen medio para monitorizar la efectividad de las actividades de la SCM de una manera continuada. Estas

mediciones son útiles para caracterizar el estado actual del proceso y para proporcionar una base para hacer comparaciones con el tiempo. Los análisis de las mediciones pueden proporcionar ideas que lleven a cambios en el proceso y a las correspondientes actualizaciones del SCMP.

Las bibliotecas de software y las diferentes habilidades de las herramientas de la SCM proporcionan fuentes para extraer información acerca de las características de los procesos de la SCM (e información del proyecto y de gestión). Por ejemplo, sería útil para evaluar los criterios usados para determinar qué niveles de autoridad son los óptimos para autorizar ciertos tipos de cambios, el tener información acerca del tiempo necesario para realizar distintos tipos de cambios.

Se debe tener cuidado en asegurarse de mantener el objetivo del seguimiento en los descubrimientos que se pueden ganar de las mediciones y no en las mediciones en si mismas. El KA del Proceso de la Ingeniería del Software presenta una discusión acerca de medidas del producto y de los procesos. El programa de medidas del software se describe en el KA de la Gestión del Ingeniería del Software.

1.5.2 Auditorías durante el proceso de la SCM [Buc96:c15]

Se pueden llevar a cabo auditorías durante el proceso de ingeniería del software para investigar el estado actual de elementos específicos de la configuración o para evaluar la implementación del proceso de la SCM. Las auditorías durante el proceso de la SCM proporcionan un mecanismo más formal para monitorizar aspectos seleccionados del proceso y se podría coordinar con la función del SQA. Véase también el punto 5 *Auditando la Configuración del Software*.

2. Identificación de la Configuración del Software [IEEE12207.0-96:c6s2.2]

La actividad de la identificación de la configuración del software identifica elementos que se han de controlar, establece métodos de identificación para los elementos y sus versiones y establece las herramientas y técnicas que se usarán para adquirir y gestionar los elementos controlados. Estas actividades proporcionan la base para las otras actividades de la SCM.

2.1 Identificando los Elementos a Controlar [Ber92:c8; IEEE828-98:c4s3.1; Pau93:L2-83; Som05:c29]

Un primer paso para controlar cambios es identificar los elementos de software a ser controlados. Esto requiere comprender la configuración del software en el contexto de la configuración del sistema, seleccionando elementos de configuración de software, desarrollando estrategias para etiquetar elementos de software y describir las relaciones entre ellos y identificar las líneas base que se usarán, además de los procedimientos de adquisición de elementos para una línea base.

2.1.1 Configuración del software [Buc96:c4; c6, Pre04:c27]

Una configuración del software es el conjunto de características funcionales y físicas del software tal y como se han definido en la documentación técnica o conseguido en un producto final (IEEE610.12-90). Se puede ver como parte de una configuración del sistema

más general.

2.1.2 Elemento de configuración del software [Buc96:c4;c6; Con98:c2; Pre04:c27]

Un elemento de configuración de software (SCI) es una agregación de software, asignado para tener gestión de configuración y se trata como una sola entidad en el proceso de la SCM (IEEE610.12-90). La SCM controla un conjunto variado de elementos aparte del código. Los planes, documentación de especificaciones y diseño, material de pruebas, herramientas de software, código fuente y ejecutable, bibliotecas de código, datos y diccionarios de datos y documentación para la instalación, mantenimiento, operación y uso del software, están entre los elementos de software con potencial para convertirse en SCIs.

La selección de SCIs es un proceso importante en el que se ha de conseguir un equilibrio entre proporcionar una visibilidad adecuada para el control del proyecto y proporcionar un número manejable de elementos a controlar. Se puede encontrar una lista con criterios para la selección de SCIs en [Ber92].

2.1.3 Relaciones entre elementos de la configuración del software [Con98:c2; Pre04:c27]

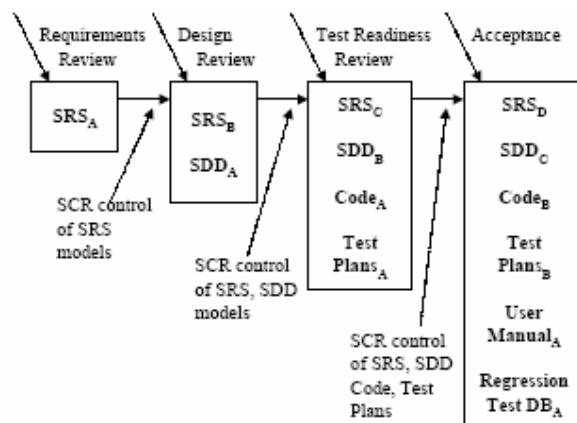
La relación estructural entre los SCIs seleccionados y sus partes constituyentes, afectan a otras actividades o tareas de la SCM, como la construcción del software o el análisis del impacto de los cambios propuestos. El seguimiento adecuado de estas relaciones es también importante como soporte a las operaciones de seguimiento. La necesidad de asignar los elementos identificados a la estructura del software y la necesidad de soportar la evolución de los elementos de software y sus relaciones, se debería considerar durante el diseño de los métodos de identificación para los SCIs.

2.1.4 Versiones del software [Bab86:c2]

Los elementos de software evolucionan al mismo tiempo que el proyecto de software avanza. Una *versión* de un elemento de software es un elemento identificado y especificado particularmente. Se puede pensar en ella como el estado de un elemento que evoluciona. [Con98:c3-c5] Una *revisión* es una nueva versión de un elemento que reemplazará la versión anterior. Una *variante* es una nueva versión de un elemento que se añadirá la configuración sin reemplazar la versión anterior.

2.1.5 Línea base [Bab86:c5; Buc96:c4; Pre04:c27]

La línea base de un software es un conjunto de elementos **Figura 4** Adquisición de elementos de configuración del software que se han designado



formalmente y fijados en un momento determinado durante el ciclo de vida del software. El término se usa también para referirse a una versión en particular de un elemento de la configuración del software acordada previamente. En cualquiera de los casos, la línea base solo se puede cambiar por medio de procedimientos de control de cambios formales. Una línea base representa, junto con todos los cambios aprobados para la línea base, la configuración actual aprobada.

Algunas líneas base comúnmente utilizadas son la funcional, la asignada, la de desarrollo y la de productos (véase por ejemplo [Ber92]). La línea base funcional se corresponde con los requerimientos del sistema ya verificados. La línea base asignada se corresponde con las especificaciones de los requerimientos del sistema y las especificaciones de los requerimientos de las interacciones entre software. La línea base de desarrollo representa la configuración evolutiva del software en momentos determinados durante el ciclo de vida del proyecto. La autoridad de cambios para dicha línea base es normalmente la responsabilidad de la organización de desarrollo, pero se podría compartir con otras organizaciones (por ejemplo la de SCM o la de Pruebas). La línea base de un producto se corresponde con el producto de software completo, entregado para integración de sistemas. Las líneas base a usar en un proyecto determinado, junto con los niveles de autoridad asociados necesarios para la aprobación de cambios, se identifican normalmente en el SCMP.

2.1.6 Adquisición de elementos de configuración del software [Buc96:c4]

Diferentes elementos de configuración del software se ponen bajo el control de la SCM en momentos distintos; lo que significa que se añaden a una línea base en particular en momentos específicos del ciclo de vida del software. El evento que da comienzo al proceso es la terminación de alguna forma de tarea formal de aceptación, como una revisión formal. La figura 2 caracteriza el crecimiento de elementos en una línea base durante el ciclo de vida. Esta figura se basa en el modelo de cascada solamente por motivos ilustrativos; los subscripts usados en la figura indican la versión de los elementos durante su evolución. La petición de cambios del software (SCR) se describe en el punto 3.1 *Petición, Evaluación y Aprobación de Cambios del Software*.

Seguidamente de la adquisición de un SCI, los cambios a dicho elemento se deben aprobar formalmente de la manera apropiada para el elemento y la línea base involucrados, como se define en el SCMP. Después de la aprobación, el elemento se incorpora en la línea base del software siguiendo el procedimiento apropiado.

2.2 Biblioteca de Software

[Bab86:c2; c5; Buc96:c4; IEEE828- 98:c4s3.1; Pau93:L2-82; Som01:c29]

Una biblioteca de software es una colección controlada de software y los documentos relacionados, y está diseñada para ayudar en el desarrollo del software, su uso y mantenimiento (IEEE610.12-90). También tiene un papel durante las actividades de gestión de lanzamientos y entrega de software. Se pueden usar varios tipos de bibliotecas de software, cada uno se corresponde con un nivel de madurez determinado del elemento de software. Por ejemplo, una biblioteca de desarrollo podría dar soporte durante la codificación y una biblioteca de soporte de proyectos podría dar soporte a las pruebas, mientras que una biblioteca

maestra se podría utilizar en el producto final. Se ha de asociar el nivel apropiado de control de la SCM (la línea base asociada y el nivel de autoridad para cambios) a cada biblioteca. La seguridad, en términos de control de acceso y medios de copia de seguridad, es un aspecto clave en la gestión de bibliotecas. Un modelo de una biblioteca de software se puede encontrar en [Ber92:c14].

La herramienta/s que se usan en cada biblioteca deben soportar los controles de la SCM que sean necesarios para dicha biblioteca, en términos de control de los SCIs y de acceso a la biblioteca. En el nivel de la biblioteca de desarrollo, esto significa la capacidad de gestión de código que dará servicio a desarrolladores, ingenieros de mantenimiento y SCM. Está enfocada a gestionar las versiones de los elementos de software al mismo tiempo que da soporte a las actividades de múltiples desarrolladores. A mayores niveles de control, el acceso es más restringido y el principal usuario el la SCM.

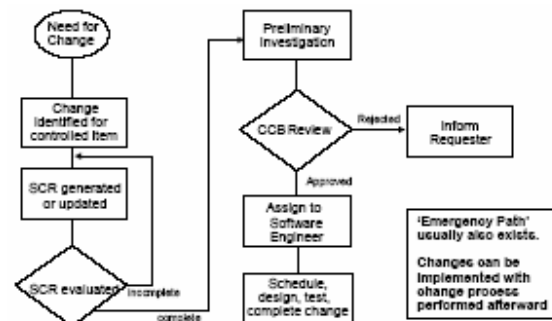
Estas bibliotecas son también una fuente importante de información para mediciones del trabajo realizado y del progreso.

3. Control de la Configuración del Software [IEEE12207.0-96:c6s2.3; Pau93:L2-84]

Al control de la configuración del software le concierne la gestión de cambios durante el ciclo de vida del software. Cubre los procesos que determinan los cambios que se realizarán, la autoridad requerida para aprobar ciertos cambios, el soporte para la implementación de dichos cambios y el concepto de desviación formal de los requerimientos del proyecto, además de las cancelaciones de requerimientos. La información derivada de estas actividades es útil para medir el tráfico de cambios y ruptura y aspectos por rehacer.

3.1 *Petición, Evaluación y Aprobación de Cambios del Software* [IEEE828-98:c4s3.2; Pre04:c27;Som05:c29]

El primer paso para gestionar cambios en elementos controlados es determinar los cambios a realizar. El proceso de petición de cambio del software (véase la Figura 5) proporciona procedimientos formales para recoger y registrar peticiones de cambios, evaluando el coste e impacto potencial de un cambio propuesto y aceptar, modificar o rechazar el cambio propuesto. Las peticiones de cambios de elementos de la configuración del software los puede originar cualquiera durante cualquier momento del ciclo de vida del software y puede incluir una solución propuesta y una prioridad. Una fuente de petición de cambios es la iniciación de



acciones **Figura 5** Flujo de un Proceso de Control de Cambios

correctivas en respuesta a los informes de problemas. El tipo de cambio (por ejemplo, un defecto o mejora) se registra normalmente en la SCR, sin importar la fuente. Esto proporciona la oportunidad de seguir defectos y recoger mediciones de la actividad de cambios por tipo de cambio. Una vez se ha recibido un SCR, se realiza una evaluación técnica (también conocida como análisis del impacto) para determinar el tamaño de las modificaciones necesarias en caso de que se aceptara la petición de cambio. Para realizar esta tarea es importante un buen entendimiento de las relaciones entre elementos de software (y posiblemente hardware). Finalmente, la evaluación de los aspectos técnicos y de gestión de la petición de cambios, será realizada por una autoridad establecida, de acuerdo con la línea base afectada, el SCI involucrado y la naturaleza del cambio y entonces se aceptará, modificará, rechazará o pospondrá el cambio propuesto.

3.1.1 Consejo de Control de la Configuración del Software [Ber92:c9; Buc96:c9,c11; Pre04:c27]

La autoridad para aceptar o rechazar los cambios propuestos, es normalmente la responsabilidad de una entidad conocida como Consejo de Control de la Configuración (CCB). En proyectos pequeños, dicha autoridad normalmente reside en el Jefe de Proyecto o algún otro individuo elegido, en vez de en un consejo de varias personas. Puede haber múltiples niveles de autoridad de cambios, dependiendo de una variedad de criterios, como cuan crítico sea el elemento involucrado, la naturaleza del cambio (por ejemplo, el impacto en el presupuesto y planificación), o el momento actual en el ciclo de vida. La composición de CCBs que se utilice para un sistema determinado varía en relación a estos criterios (siempre atendería un representante de la SCM). Cuando el alcance de la autoridad de un CCB es está limitado solamente al software, se le conoce como Consejo de Control de Configuración del Software (CCBS). Las actividades del CCB están sujetas normalmente a auditorías de la calidad de software o revisiones.

3.1.2 Proceso de petición de cambios del software [Buc96:c9,c11; Pre04:c27]

Un proceso efectivo de petición de cambio del software (SCR) requiere el uso de herramientas de soporte y procedimientos, desde formularios de papel y un procedimientos documentado hasta la herramienta electrónica para generar peticiones de cambio, imponiendo el flujo del proceso de cambios, capturando las decisiones del CCB y produciendo información del proceso de cambio. Un enlace entre las habilidades de esta herramienta y el sistema de informe de errores puede facilitar el seguimiento de soluciones para los informes de errores. Las descripciones del proceso de cambios y los formularios de soporte (información) aparecen en gran número de las referencias, por ejemplo [Ber92:c9].

3.2 Implementando Cambios en el Software [Bab86:c6; Ber92:c9; Buc96:c9,c11; IEEE828-98:c4s3.2.4; Pre04:c27; Som05:c29]

Las PCBs aprobadas se implementan utilizando los procedimientos de software definidos, de acuerdo con los requerimientos de planificación aplicables. Como se podría implementar simultáneamente un número de PCBs, es necesario proporcionar los medios para seguir que PCBs se añaden a que versiones de software y líneas bases particulares. Como parte de la finalización del proceso de cambios, los cambios completados podrían sufrir auditorías de configuración y verificación de la calidad del software. Esto incluye asegurarse de que solo se han realizado los cambios aprobados. El proceso de petición de cambios mencionado anteriormente, añadirá la información de la aprobación para el cambio a la documentación de la SCM (y otras).

La implementación real de un cambio está soportada por las habilidades de la herramienta de bibliotecas, que proporciona gestión de versiones y soporte para el almacenamiento de código. Estas herramientas proporcionan como mínimo habilidades para llevar a cabo el control de de las versiones asociadas. Herramientas más potentes pueden dar soporte al desarrollo en paralelo y entornos geográficamente distribuidos. Estas herramientas podrían aparecer como aplicaciones especializadas separadas, bajo el control de un grupo independiente de la SCM. También podrían aparecer integradas como parte del entorno de la ingeniería del software. Finalmente, podrían ser tan elementales como un sistema de control de cambios rudimentario proporcionado por el sistema operativo.

3.3 Desviaciones y Remisiones [Ber92:c9; Buc96:c12]

Las limitaciones impuestas al esfuerzo de la ingeniería del software o las especificaciones producidas durante las actividades de desarrollo podrían contener necesidades que no pueden ser satisfechas en el punto designado del ciclo de vida. Una remisión es la autorización para abandonar una necesidad antes del desarrollo del elemento. Un rechazo es la autorización para utilizar un elemento, después de su desarrollo, que se aleja de la necesidad de alguna manera. En estos casos se usa un procedimiento formal para ganar la aprobación para la desviación o remisión de las necesidades.

4. Registro del Estado de la Configuración del Software

[IEEE12207.0-96:c6s2.4; Pau93:L2-85; Pre04:c27;Som05:c29]

La contabilidad del estado de la configuración del software (SCSA) es la actividad de registrar y proporcionar la información necesaria para una gestión efectiva de la configuración del software

4.1 Información del Estado de la Configuración del Software [Buc96:c13; IEEE828-98:c4s3.3]

La actividad de la SCSA diseña y opera un sistema para la captura y generación de los informes necesarios durante el ciclo de vida. Como en cualquier sistema de información, se debe identificar, recoger y mantener la información del estado de la configuración que se ha de

1 gestionar según las configuraciones evolucionan. Se
2 necesitan varias mediciones e información para dar
3 soporte al proceso de la SCM y para cubrir las
4 necesidades de informes del estado de la configuración
5 de las actividades de gestión, ingeniería del software y
6 otras actividades relacionadas. Los tipos de información
7 disponible incluyen la identificación de la configuración
8 aprobada y la identificación y estado de implementación
9 actual de cambios, desviaciones y remisiones. Se puede
10 encontrar una lista parcial con elementos de datos
11 importantes en [Ber92:c10].
12 Es necesario algún tipo de soporte de herramientas
13 automáticas para llevar a cabo las tareas de recogida de
14 datos y generación de informes de la SCSA. Podría ser
15 una habilidad de la base de datos o una herramienta
16 independiente o la habilidad del entorno de una
17 herramienta integrada más grande.

18 4.2 Informes del Estado de la Configuración del 19 Software [Ber92:c10; Buc96:c13]

20 Los informes generados pueden ser usados por varios
21 elementos de la organización o del proyecto, incluyendo
22 el equipo de desarrollo, el equipo de mantenimiento, la
23 gestión del proyecto y las actividades de calidad de
24 software. Los informes pueden tener la forma de
25 respuestas inmediatas a preguntas específicas o ser
26 informes prediseñados producidos periódicamente.
27 Alguna de la información producida por las actividades
28 de contabilidad del estado durante el curso del ciclo de
29 vida podría acabar siendo registros de la garantía de la
30 calidad.
31 Además de informar del estado actual de la
32 configuración, la información obtenida por la SCSA
33 puede usarse como base para varias mediciones útiles
34 para la gestión, desarrollo y SCM. Un ejemplo podría
35 ser el número de cambios pedidos por ECS y el tiempo
36 medio necesario para implementar una petición de
37 cambio.

38 5. Auditoría de la Configuración del Software 39 [IEEE828-98:c4s3.4; IEEE12207.0- 40 96:c6s2.5; Pau93:L2-86; Pre04:c26c27]

41 La auditoría de software es una actividad que se realiza
42 para evaluar independientemente la conformidad de
43 productos de software y procesos con las regulaciones,
44 estándares, guías, planes y procedimientos (IEEE1028-
45 97). Las auditorías se llevan a cabo de acuerdo con un
46 proceso bien definido que consiste en varias
47 responsabilidades y papeles de auditoría. En
48 consecuencia, cada auditoría se debe planear con
49 cuidado. Una auditoría requiere un número de personas
50 que realizarán una variedad de tareas en un periodo de
51 tiempo bastante reducido. Herramientas que den soporte
52 a la planificación y ejecución de la auditoría pueden
53 facilitar el proceso enormemente. Se pueden encontrar
54 consejos para realizar auditorías de software en varias
55 referencias, como [Ber92:c11, Buc96:c15] y
56 (IEEE1028-97).
57 La actividad de auditoría de la configuración del
58 software determina el grado en que un elemento
59 satisface las características funcionales y físicas. Se
60 pueden realizar auditorías informales de este tipo en
61 momentos clave del ciclo de vida. Hay dos tipos de
62 auditorías que podrían ser requeridas por el contrato (por
63 ejemplo, en contratos para software crítico): la Auditoría
64 de la Configuración Funcional (FCA) y la Auditoría de
65 la Configuración Física (PCA). El completar con éxito
66 estas auditorías puede ser un prerequisite para

67 establecer la línea base del producto. Buckley
68 [Buc96:c15] contrasta los objetivos de las FCA y PCA
69 en los contextos de software y hardware y recomienda
70 que se evalúe cuidadosamente la necesidad de una FCA
71 y PCA de software antes de realizarlas.

72 5.1 Auditoría de la Configuración Funcional del 73 Software

74 El propósito de la FCA del software es asegurarse de
75 que el elemento de software que se audita es consistente
76 con la especificación. Los resultados de la verificación y
77 validación del software son actividades clave como
78 entrada de datos para esta auditoría.

79 5.2 Auditoría de la Configuración Física del Software

80 El propósito de la auditoría de la configuración física del
81 software (PCA) es asegurarse de que el diseño y la
82 documentación de referencia son consistentes con el
83 producto de software tal y como se ha construido.

84 5.3 Auditorías durante el proceso de una Línea Base de 85 Software

86 Tal y como se menciona anteriormente, las auditorías se
87 pueden llevar a cabo durante el proceso de desarrollo
88 para investigar el estado actual de un elemento de la
89 configuración específico. En dicho caso, se podría
90 aplicar una auditoría a elementos seleccionados de la
91 línea base para asegurarse de que el rendimiento es
92 consistente con las especificaciones o para asegurarse de
93 que la documentación continua siendo consistente con el
94 elemento de la línea base que se está desarrollando.

95 6. Gestión del Lanzamiento y Distribución del 96 Software [IEEE12207.0-96:c6s2.6]

97 El término “lanzamiento” se usa en este contexto para
98 referirse a la distribución un elemento de la
99 configuración del software fuera de la actividad de
100 desarrollo. Esto incluye tanto lanzamientos internos
101 como la distribución a clientes. Cuando una versión
102 diferente de un elemento de software está disponible
103 para ser entregada, como las versiones para diferentes
104 plataformas o versiones con diferentes capacidades, es
105 normalmente necesario preparar una versión específica y
106 empaquetar los materiales adecuados para distribuirla.
107 La biblioteca de software es un elemento clave para
108 realizar las tareas de lanzamiento y distribución.

109 6.1 Construcción del Software [Bab86:c6; Som05:c29]

110 La construcción del software es la actividad de combinar
111 la versión correcta de los elementos de configuración del
112 software, usando la configuración de datos apropiada, en
113 un programa ejecutable para su distribución a los
114 clientes u otros receptores, como la actividad de pruebas.
115 Las instrucciones de construcción se aseguran de que se
116 toman los pasos de construcción adecuados y en la
117 secuencia correcta. Además de construir software para
118 un nuevo lanzamiento, la SCM normalmente necesita
119 ser capaz de reproducir lanzamientos previos para
120 recuperación, pruebas, mantenimiento u otros propósitos
121 de lanzamiento adicionales.
122 El software se construye usando versiones particulares
123 de la herramientas de soporte, como compiladores.
124 Podría ser necesario reconstruir una copia exacta de un
125 elemento de configuración que se haya construido

1 previamente. En ese caso, las herramientas de soporte y
2 las instrucciones de construcción asociadas deben de
3 estar bajo el control de la SCM para asegurarse de la
4 disponibilidad de la las versiones correctas de las
5 herramientas.

6 Las habilidades de una herramienta son útiles para
7 seleccionar la versión correcta de elementos de software
8 para un entorno destino determinado y para el proceso
9 de construir el software automáticamente con las
10 versiones seleccionadas y los datos de configuración
11 apropiados. En proyectos grandes con desarrollo en
12 paralelo o en entornos de desarrollo distribuido, estas
13 habilidades de las herramientas son necesarias. La
14 mayoría de los entornos de ingeniería del software
15 proporcionan esta habilidad. Estas herramientas varían
16 en complejidad, desde las que requieren que el ingeniero
17 de software aprenda un lenguaje de guiones específico a
18 soluciones gráficas que ocultan la mayor parte de la
19 complejidad en una solución de construcción
20 “inteligente”.

21 El proceso y los productos de la construcción están
22 sujetos, normalmente, a verificación de la calidad del
23 software. Los resultados de un proceso de construcción
24 se podrían necesitar para futuras referencias y podrían
25 convertirse en registros de la garantía del software.

26 6.2 Gestión del Lanzamiento del Software [Som05:c29]

27 La gestión de lanzamiento del software conlleva la
28 identificación, empaquetamiento y distribución de los
29 elementos de un producto, por ejemplo, programas
30 ejecutables, documentación, notas de lanzamiento y
31 datos de configuración. Dado que los cambios pueden
32 ocurrir constantemente, una de las preocupaciones en la

33 gestión del lanzamientos es determinar cuando realizar
34 un lanzamiento. La severidad de los problemas
35 solucionados por el lanzamiento afecta a esta decisión
36 (Som01). La tarea de empaquetamiento debe identificar
37 que elementos del producto se deben distribuir y por
38 tanto seleccionar las variantes correctas de dichos
39 elementos, dada la aplicación que se le quiere dar al
40 producto. La información que documenta el contenido
41 físico del lanzamiento se conoce como documento de
42 descripción de la versión. Las notas del lanzamiento
43 normalmente describen nuevas habilidades, problemas
44 conocidos y requisitos necesarios de la plataforma para
45 la operación adecuada del producto. El paquete que se
46 distribuirá también contiene instrucciones de instalación
47 o actualización. El último se puede complicar porque
48 algunos usuarios podrían tener productos que son
49 antiguos por varias versiones. Finalmente, en algunos
50 casos, se podrían requerir la actividad de gestión de
51 lanzamientos para el seguimiento de la distribución del
52 producto a varios clientes o sistemas objetivo. Un
53 ejemplo sería el caso en el que se requiriese que un
54 proveedor tiene que notificar a un cliente de nuevos
55 problemas.

56 Las habilidades de una herramienta son necesarias para
57 dar soporte a estas funciones de gestión de los
58 lanzamientos. Es útil tener una conexión con las
59 habilidades de la herramienta para dar soporte a los
60 procesos de peticiones de cambios, de tal forma que se
61 puedan relacionar los contenidos de un lanzamiento con
62 los SCR que se han recibido. Esta habilidad de las
63 herramientas también podría mantener información en
64 varias plataformas destino y de varios entornos de
65 clientes.

1 REFERENCIAS RECOMENDADAS PARA SCM

2
3 [Bab86] W.A. Babich, *Software Configuration*
4 *Management, Coordination for Team Productivity*,
5 Addison-Wesley, 1986.

6
7 [Ber92] H.R. Berlack, *Software Configuration*
8 *Management*, John Wiley & Sons, 1992.

9
10 [Buc96] F.J. Buckley, *Implementing Configuration*
11 *Management: Hardware, Software, and Firmware*,
12 second ed., IEEE Computer Society Press, 1996.

13
14 [Con98] R. Conradi and B. Westfechtel, "Version Models
15 for Software Configuration Management," *ACM*
16 *Computing Surveys*, vol. 30, iss. 2, June 1998.

17
18 [Dar90] S.A. Dart, *Spectrum of Functionality in*
19 *Configuration Management Systems*, Software
20 Engineering Institute, Carnegie Mellon University, 1990.

21
22 [IEEE828-98] IEEE Std 828-1998, *IEEE Standard for*
23 *Software Configuration Management Plans*, IEEE, 1998.

24 [IEEE12207.0-96] IEEE/EIA 12207.0-1996//ISO/
25 IEC12207:1995, *Industry Implementation of Int. Std.*
26 *ISO/IEC 12207:95, Standard for Information Technology-*
27 *Software Life Cycle Processes*, IEEE, 1996.

28
29 [Mid97] A.K. Midha, "Software Configuration
30 Management for the 21st Century," *Bell Labs Technical*
31 *Journal*, vol. 2, iss. 1, Winter 1997, pp. 154-165.

32
33 [Moo98] J.W. Moore, *Software Engineering Standards: A*
34 *User's Roadmap*, IEEE Computer Society, 1998.

35
36 [Pau93] M.C. Paulk et al., "Key Practices of the
37 Capability Maturity Model, Version 1.1," technical report
38 CMU/SEI-93-TR-025, Software Engineering Institute,
39 Carnegie Mellon University, 1993.

40
41 [Pre04] R.S. Pressman, *Software Engineering: A*
42 *Practitioner's Approach*, Sixth ed, McGraw-Hill, 2004.

43
44 [Roy98] W. Royce, *Software Project Management, A*
45 *United Framework*, Addison-Wesley, 1998.

46
47 [Som05] I. Sommerville, *Software Engineering*, seventh
48 ed., Addison-Wesley, 2005.

**APÉNDICE A. LISTA DE LECTURAS
ADICIONALES**

(Bab86) W.A. Babich, *Software Configuration Management, Coordination for Team Productivity*, Addison-Wesley, 1986.

(Ber92) H.R. Berlack, *Software Configuration Management*, John Wiley & Sons, 1992.

(Ber97) E.H. Bersoff, "Elements of Software Configuration Management," in *Software Engineering*, M. Dorfman and R.H. Thayer, eds., IEEE Computer Society Press, 1997.

(Buc96) F.J. Buckley, *Implementing Configuration Management: Hardware, Software, and Firmware*, second ed., IEEE Computer Society Press, 1996.

(EIE98) K. El-Emam et al., "SPICE, The Theory and Practice of Software Process Improvement and Capability Determination," presented at IEEE Computer Society, 1998.

(Est95) J. Estublier, "Software Configuration Management," presented at ICSE SCM-4 and SCM-5 Workshops, Berlin, 1995.

(Gra92) R.B. Grady, *Practical Software Metrics for Project Management and Process Management*, Prentice Hall, 1992.

(Hoe02) A. v. d. Hoek, "Configuration Management Yellow Pages," 2002, available at http://www.cmtoday.com/yp/configuration_management.html.

(Hum89) W. Humphrey, *Managing the Software Process*, Addison-Wesley, 1989.

(Pau95) M.C. Paulk et al., *The Capability Maturity Model, Guidelines for Improving the Software Process*, Addison- Wesley, 1995.

(Som01a) I. Sommerville, "Software Configuration Management," presented at ICSE SCM-6 Workshop, Berlin, 2001.

(USNRC1.169-97) USNRC Regulatory Guide 1.169, "Configuration Management Plans for Digital Computer Software Used in Safety Systems of Nuclear Power Plants," presented at U.S. Nuclear Regulatory Commission, Washington, D.C., 1997.

(Vin88) J. Vincent, A. Waters, and J. Sinclair, *Software Quality Assurance: Practice and Implementation*, Prentice Hall, 1988.

(Whi91) D. Whitgift, *Methods and Tools for Software Configuration Management*, John Wiley & Sons, 1991.

APÉNDICE B. LISTA DE ESTÁNDARES

(IEEE730-02) IEEE Std 730-2002, *IEEE Standard for Software Quality Assurance Plans*, IEEE, 2002.

(IEEE828-98) IEEE Std 828-1998, *IEEE Standard for Software Configuration Management Plans*, IEEE, 1998.

(IEEE1028-97) IEEE Std 1028-1997 (R2002), *IEEE Standard for Software Reviews*, IEEE, 1997.

(IEEE12207.0-96) IEEE/EIA 12207.0-1996//ISO/IEC12207:1995, *Industry Implementation of Int. Std. ISO/IEC 12207:95, Standard for Information Technology-Software Life Cycle Processes*, IEEE, 1996.

(IEEE12207.1-96) IEEE/EIA 12207.1-1996, *Industry Implementation of Int. Std. ISO/IEC 12207:95, Standard for Information Technology-Software Life Cycle Processes - Life Cycle Data*, IEEE, 1996.

(IEEE12207.2-97) IEEE/EIA 12207.2-1997, *Industry Implementation of Int. Std. ISO/IEC 12207:95, Standard for Information Technology-Software Life Cycle Processes -Implementation Considerations*, IEEE, 1997.

(ISO15846-98) ISO/IEC TR 15846:1998, *Information Technology - Software Life Cycle Processes - Configuration Management*, ISO and IEC, 1998.