

# ADMINISTRACIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

## CONCEPTOS CLAVE

administración de contenido .	517
auditoría de configuración . .	514
líneas de referencia . . . . .	504
control de cambio . . . . .	511
control de versión . . . . .	510
identificación . . . . .	509
ítems de configuración de software (ICS) . . . . .	505
objeto de configuración . . . .	505
objetos de configuración de webapps . . . . .	517
proceso ACS . . . . .	508
reporte de estado . . . . .	615
repositorio . . . . .	506
webapps . . . . .	515

Cuando se construye software de computadoras, el cambio es inevitable. Y el cambio aumenta el nivel de confusión cuando los miembros de un equipo de software trabajan en un proyecto. La confusión surge cuando los cambios no se analizan antes de que se realicen, cuando no se registran antes de que se implanten, si no se reportan a quienes tienen necesidad de conocerlos o si no se controlan en forma que mejore la calidad y se reduzca el error. Babich [Bab86] analiza esto cuando afirma:

El arte de coordinar el desarrollo de software para minimizar [...] la confusión se llama administración de la configuración, que es el arte de identificar, organizar y controlar las modificaciones que se hacen al software que construirá un equipo de programación. La meta es maximizar la productividad al minimizar los errores.

La administración de la configuración del software (ACS) es una actividad sombrilla que se aplica a lo largo del proceso de software. Puesto que el cambio puede ocurrir en cualquier momento, se desarrollan actividades ACS para 1) identificar el cambio, 2) controlar el cambio, 3) garantizar que el cambio se implementó de manera adecuada y 4) reportar los cambios a otros que puedan estar interesados.

Es importante hacer una distinción clara entre el apoyo al software y la administración de la configuración del software. El apoyo es un conjunto de actividades de ingeniería de software que ocurren después de que éste se entregó al cliente y de que se puso en operación. La administración de la configuración del software es un conjunto de actividades de rastreo y control que inicia cuando comienza un proyecto de ingeniería de software y sólo termina cuando el software se retira de la operación.

## UNA MIRADA RÁPIDA

**¿Qué es?** Cuando se construye software de computadora, ocurren cambios. Y puesto que ocurren, es necesario administrarlos de manera efectiva. La administración de la configuración del software (ACS), también llamada gestión del cambio, es un conjunto de actividades diseñadas para administrar el cambio mediante la identificación de los productos de trabajo que es probable que cambien, el establecimiento de relaciones entre ellos, la definición de mecanismos para administrar diferentes versiones de dichos productos de trabajo y el control de los cambios impuestos, así como la auditoría y reporte de los cambios realizados.

**¿Quién lo hace?** Todos los involucrados en el proceso de software se relacionan en cierta medida con la gestión del cambio, pero en ocasiones se crean posiciones de apoyo especializadas para administrar el proceso ACS.

**¿Por qué es importante?** Si no se controla el cambio, éste lo controla a uno. Y eso nunca es bueno. Es muy fácil que un torrente de cambios descontrolados convierta en caos un proyecto de software bien estructurado. Como consecuencia, la calidad del software se reduce y la entre-

ga se demora. Por dicha razón, la gestión del cambio es parte esencial de la administración de la calidad.

**¿Cuáles son los pasos?** Puesto que muchos productos de trabajo se realizan cuando el software se construye, cada uno debe identificarse de manera única. Una vez logrado, pueden establecerse mecanismos para control de versión y de cambio. Para garantizar que la calidad se mantiene conforme se realizan cambios, audite el proceso; y para asegurarse que quienes deben conocerlos estén informados acerca de los cambios, realice reportes.

**¿Cuál es el producto final?** Un Plan de Administración de la Configuración del Software define la estrategia del proyecto para la gestión del cambio. Además, cuando se invoca ACS formal, el proceso de control de cambio produce solicitudes de cambio de software, reportes y órdenes de cambio de ingeniería.

**¿Cómo me aseguro de que lo hice bien?** Cuando todo producto de trabajo pueda explicarse, rastrearse y controlarse; cuando todo cambio pueda rastrearse y analizarse; cuando todos los que deben saber acerca de un cambio estén informados, entonces la gestión del cambio se hizo correctamente.

Una meta principal de la ingeniería de software es mejorar la facilidad con la que los cambios pueden acomodarse y reducir la cantidad de esfuerzo empleado cuando deban realizarse cambios. En este capítulo se estudian las actividades específicas que permiten administrar el cambio.

## 22.1 ADMINISTRACIÓN DE LA CONFIGURACIÓN DEL SOFTWARE

La salida del proceso de software es información que puede dividirse en tres categorías amplias: 1) programas de cómputo (tanto en el nivel de fuente como en formatos ejecutables), 2) productos de trabajo que describen los programas de cómputo (dirigidos a varios participantes) y 3) datos o contenido (incluidos dentro del programa o externos a él). Los ítems que comprenden toda la información producida como parte del proceso de software se llaman colectivamente *configuración del software*.

### Cita:

"No hay nada permanente, excepto el cambio."

Heráclito, 500 a.C.

Conforme avanza el trabajo de ingeniería de software, se crea una jerarquía de *ítems de configuración del software* (ICS): un elemento de información nominado que puede ser tan pequeño como un solo diagrama UML o tan grande como el documento de diseño completo. Si cada ICS simplemente conduce a otros ICS, dará como resultado poca confusión. Por desgracia, en el proceso entra otra variable: *el cambio*, que puede ocurrir en cualquier momento, por cualquier razón. De hecho, la *Primera Ley de la Ingeniería de Sistemas* [Ber80] establece: "Sin importar dónde se esté en el ciclo de vida del sistema, el sistema cambiará, y el deseo por cambiar persistirá a lo largo del ciclo de vida."

¿Cuál es el origen de estos cambios? La respuesta a esta pregunta es tan variada como los mismos cambios. Sin embargo, existen cuatro fuentes fundamentales de cambio:

- Nuevas condiciones empresariales o de mercado dictan los cambios en los requerimientos del producto o en las reglas empresariales.
- Nuevas necesidades de los accionistas demandan modificación a los datos producidos por los sistemas de información, a la funcionalidad que entregan los productos o a los servicios que ofrece un sistema basado en computadora.
- La reorganización o crecimiento/reducción de la empresa produce cambios en las prioridades proyectadas o en la estructura del equipo de ingeniería de software.
- Restricciones presupuestales o de calendario causan una redefinición del sistema o del producto.

La administración de la configuración del software es un conjunto de actividades que se desarrollaron para administrar el cambio a lo largo del ciclo de vida del software de computadora. La ACS puede verse como una actividad que garantiza la calidad del software y que se aplica a lo largo del proceso de software. En las siguientes secciones se describen las principales tareas ACS y los conceptos importantes que pueden ayudar a gestionar el cambio.

### 22.1.1 Un escenario ACS<sup>1</sup>

Un escenario operativo de administración del cambio (AC) típico involucra a un gerente de proyecto que está a cargo de un grupo de software, a un gerente de configuración responsable de los procedimientos y políticas AC, a los ingenieros de software encargados de desarrollar y mantener el producto de software y al cliente que usa el producto. En el escenario, se supone que el producto es pequeño y que involucra a alrededor de 15 000 líneas de código desarrollado por un equipo de seis personas. (Observe que es posible que existan otros escenarios de equipos

<sup>1</sup> Esta sección se extrajo de [Dar01]. El permiso especial para reproducir "Spectrum of functionality in CM System", de Susan Dart [Dar01], © 2001 de Carnegie Mellon University, se obtuvo del Software Engineering Institute.

**?** ¿Cuáles son las metas de y las actividades realizadas por cada uno de los elementos constituyentes involucrados en la administración del cambio?

### **PUNTO CLAVE**

Debe existir un mecanismo para asegurar que los cambios simultáneos hechos al mismo componente se rastreen, gestionen y ejecuten de manera adecuada.

más pequeños o más grandes, pero, en esencia, hay temas genéricos que cada uno de estos proyectos enfrenta con respecto a la AC).

En el nivel operativo, el escenario involucra varios roles y tareas. Para el gerente de proyecto, la meta es garantizar que el producto se desarrolla dentro de cierto marco temporal. Por tanto, monitorea el progreso del desarrollo y reconoce y reacciona ante los problemas. Esto se hace mediante la generación y el análisis de reportes acerca del estado del sistema de software y al realizar revisiones al sistema.

Las metas del gerente de configuración son garantizar que se sigan los procedimientos y políticas para crear, cambiar y probar el código, así como hacer accesible la información acerca del proyecto. Para implantar técnicas a fin de mantener el control sobre los cambios de código, este gerente introduce mecanismos para: realizar peticiones oficiales de cambios, evaluarlos (mediante un Consejo de Control de Cambios que sea responsable de aprobar los cambios al sistema de software) y autorizarlos. El gerente elabora y difunde la lista de tareas para los ingenieros y básicamente crea el contexto del proyecto. Además, recopila estadísticas acerca de los componentes que hay en el sistema de software, tales como la información que determina cuáles componentes del sistema son problemáticos.

Para los ingenieros de software, la meta es trabajar eficazmente. Esto significa que los ingenieros no deben interferir innecesariamente unos con otros en la creación y prueba del código y en la producción de productos operativos de apoyo. Pero, al mismo tiempo, deben intentar comunicarse y coordinarse de manera eficiente. Específicamente, los ingenieros usan herramientas que ayudan a construir un producto de software consistente. Se comunican y coordinan al notificarse unos con otros las tareas requeridas y las tareas completadas. Los cambios se propagan a través del trabajo mutuo mediante fusión de archivos. Existen mecanismos para garantizar que, para componentes que experimentan cambios simultáneos, hay alguna forma de resolver los conflictos y la fusión de cambios. Se conserva una historia de la evolución de todos los componentes del sistema, una bitácora con las razones de los cambios y un registro de lo que realmente cambió. Los ingenieros tienen su propio espacio de trabajo para crear, cambiar, poner a prueba e integrar código. En cierto punto, el código se convierte en una línea de referencia desde la cual continúan mayores desarrollos y se realizan variantes para otras máquinas objetivo.

El cliente usa el producto. Puesto que éste se encuentra bajo control AC, el cliente sigue procedimientos formales para solicitar cambios y para indicar errores en el producto.

De manera ideal, un sistema AC utilizado en este escenario debe apoyar todos estos roles y tareas, es decir, los roles determinan la funcionalidad requerida de un sistema AC. El gerente de proyecto ve la AC como un mecanismo de auditoría; el gerente de configuración la considera un mecanismo de control, rastreo y generación de políticas; el ingeniero de software, como un mecanismo de control de cambio, construcción y acceso; y el cliente la ve como un camino para garantizar la calidad.

### **22.1.2 Elementos de un sistema de administración de la configuración**

En su exhaustivo artículo acerca de la administración de la configuración del software, Susan Dart [Dar01] identifica cuatro elementos importantes que deben existir cuando se desarrolla un sistema de administración de la configuración:

- *Elementos componentes:* conjunto de herramientas acopladas dentro de un sistema de administración de archivos (por ejemplo, base de datos) que permite el acceso a cada ítem de configuración del software, así como su gestión.
- *Elementos de proceso:* colección de acciones y tareas que definen un enfoque efectivo de la gestión del cambio (y actividades relacionadas) para todos los elementos constituyentes involucrados en la administración, ingeniería y uso del software.

- *Elementos de construcción:* conjunto de herramientas que automatizan la construcción de software al asegurarse de que se ensambló el conjunto adecuado de componentes validados (es decir, la versión correcta).
- *Elementos humanos:* conjunto de herramientas y características de proceso (que abarcan otros elementos AC) utilizados por el equipo de software para implementar ACS efectiva.

Estos elementos (que se estudiarán con más detalle en secciones posteriores) no son mutuamente excluyentes. Por ejemplo, los elementos componentes trabajan en conjunción con los elementos de construcción conforme evoluciona el proceso de software. Los elementos de proceso guían muchas actividades humanas que se relacionan con la ACS y, por tanto, también pueden considerarse como elementos humanos.

### 22.1.3 Líneas de referencia



*La mayoría de los cambios de software se justifican, así que no hay razón para quejarse por su presencia. En su lugar, asegúrese de que tiene los mecanismos para lidiar con ellos.*

El cambio es un hecho de vida en el desarrollo de software. Los clientes quieren modificar los requerimientos. Los desarrolladores quieren cambiar el enfoque técnico. Los gerentes quieren modificar la estrategia del proyecto. ¿Por qué todas estas modificaciones? La respuesta realmente es muy simple. Conforme pasa el tiempo, todos los elementos constituyentes saben más (acerca de lo que necesitan, sobre qué enfoque sería mejor, cómo realizarlo y todavía obtener dinero). Este conocimiento adicional es la fuerza motora que hay detrás de la mayoría de los cambios y que conduce a un enunciado de hechos que es difícil de aceptar para muchos profesionales de la ingeniería de software: *¡la mayoría de los cambios se justifican!*

Una *línea de referencia* es un concepto de administración de la configuración del software que le ayuda a controlar el cambio sin impedir seriamente cambios justificados. El IEEE (IEEE Std. No. 610.12-1990) define una línea de referencia como:

Una especificación o producto que se revisó formalmente y con el que se estuvo de acuerdo, que a partir de entonces sirve como base para un mayor desarrollo y que puede cambiar sólo a través de procedimientos de control de cambio formal.

Antes de que un ítem de configuración del software se convierta en línea de referencia, los cambios pueden realizarse rápida e informalmente. No obstante, una vez establecida la línea de referencia, pueden realizarse cambios, pero debe aplicarse un procedimiento formal específico para evaluar y verificar cada uno de ellos.

En el contexto de la ingeniería de software, una línea de referencia es un hito en el desarrollo del software. Una línea de referencia se marca al entregar uno o más ítems de configuración del software que se aprobaron como consecuencia de una revisión técnica (capítulo 15). Por ejemplo, los elementos de un modelo de diseño se documentaron y revisaron. Se encontraron y corrigieron errores. Una vez que todas las partes del modelo se revisaron, corrigieron y luego aprobaron, el modelo de diseño se convierte en línea de referencia. Los cambios adicionales a la arquitectura del programa (documentada en el modelo de diseño) pueden realizarse sólo después de que cada uno se evalúa y aprueba. Aunque las líneas de referencia pueden definirse en cualquier nivel de detalle, en la figura 22.1 se muestran las líneas de referencia de software más comunes.

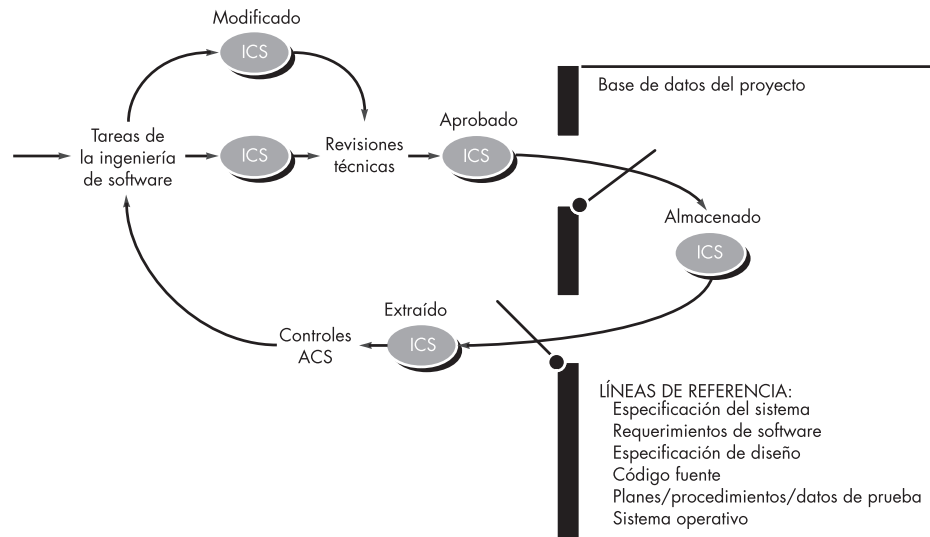


*Asegúrese de que la base de datos del proyecto se mantenga en una ubicación centralizada controlada.*

En la figura 22.1 también se ilustra la progresión de eventos que conducen a una línea de referencia. Las tareas de la ingeniería de software producen uno o más ICS. Después de revisar y aprobar los ICS, se colocan en una *base de datos del proyecto* (también llamada *librería de proyecto* o *repositorio de software*, y que se estudia en la sección 22.2). Cuando un miembro de un equipo de ingeniería de software quiere hacer una modificación a un ICS que se ha convertido en línea de referencia, se copia de la base de datos del proyecto en el espacio de trabajo privado del ingeniero. Sin embargo, este ICS extraído puede modificarse solamente si se siguen controles ACS (que se estudian más adelante en este capítulo). Las flechas de la figura 22.1 ilustran la ruta de modificación de un ICS convertido en línea de referencia.

FIGURA 22.1

ICS como línea de referencia y base de datos del proyecto



#### 22.1.4 Ítems de configuración del software

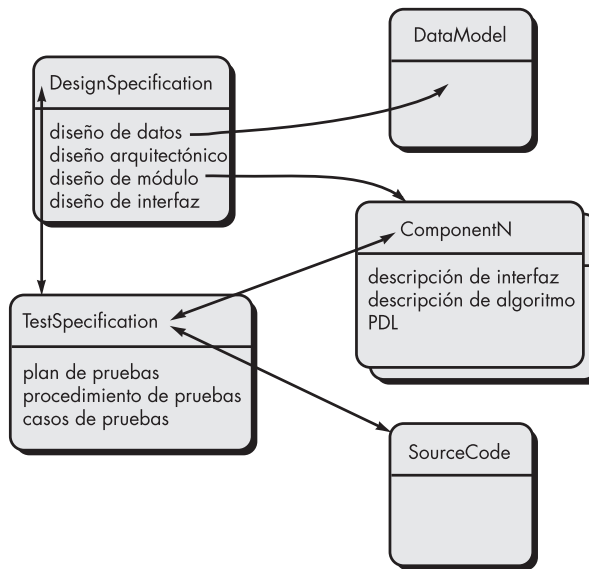
Ya se definió un ítem de configuración del software como la información que se crea como parte del proceso de ingeniería de software. En última instancia, un ICS podría considerarse como una sola sección de una gran especificación o como un caso de prueba en una gran suite de pruebas. De manera más realista, un ICS es todo o parte de un producto de trabajo (por ejemplo, un documento, toda una suite de casos de prueba o un componente de programa nominado).

Además de los ICS que se derivan de los productos de trabajo de software, muchas organizaciones de ingeniería de software también colocan las herramientas de software bajo control de configuración, es decir, versiones específicas de editores, compiladores, navegadores y otras herramientas automatizadas se “congelan” como parte de la configuración del software. Puesto que dichas herramientas se usaron para producir documentación, código fuente y datos, deben estar disponibles cuando tengan que realizarse cambios a la configuración del software. Aunque los problemas son raros, es posible que una nueva versión de una herramienta (por ejemplo, un compilador) pueda producir resultados diferentes que la versión original. Por esta razón, las herramientas, como el software que ayudan a producir, pueden convertirse en líneas de referencia como parte de un proceso amplio de administración de la configuración.

En realidad, los ICS se organizan para formar objetos de configuración que puedan catalogarse con un solo nombre en la base de datos del proyecto. Un *objeto de configuración* tiene un nombre y atributos, y está “conectado” con otros objetos mediante relaciones. En la figura 22.2, los objetos de configuración **DesignSpecification** (especificación de diseño), **DataModel** (modelo de datos), **ComponentN** (componente n), **SourceCode** (código fuente) y **TestSpecification** (especificación de prueba) se definen cada uno por separado. Sin embargo, cada uno de los objetos se relaciona con los demás, como se muestra mediante las flechas. Una flecha curva indica una relación composicional, es decir, **DataModel** y **ComponentN** son parte del objeto **DesignSpecification**. Una flecha con doble punta indica una interrelación. Si se realizara un cambio al objeto **SourceCode**, las interrelaciones permiten determinar qué otros objetos (e ICS) pueden resultar afectados.<sup>2</sup>

<sup>2</sup> Esas relaciones se definen dentro de la base de datos. La estructura de la base de datos (repositorio) se estudia con mayor detalle en la sección 22.2.



**FIGURA 22.2****Objetos de configuración**

## 22.2 EL REPOSITORIO ACS

En los primeros días de la ingeniería de software, los ítems de configuración del software se mantenían como documentos en papel (¡o tarjetas perforadas!), colocadas en carpetas de papel o de anillos, y se almacenaban en archiveros metálicos. Este mecanismo era problemático por muchas razones: 1) con frecuencia era difícil encontrar un ítem de configuración cuando se necesitaba, 2) era muy desafiante determinar cuáles ítems cambiaban, cuándo y por quién, 3) construir una nueva versión de un programa existente consumía mucho tiempo y era proclive al error y 4) describir relaciones detalladas y complejas entre los ítems de configuración era virtualmente imposible.

En la actualidad, los ICS se mantienen en una base de datos del proyecto, o repositorio. El *Diccionario Webster* define la palabra *repositorio* como “cualquier cosa o persona que se considera como centro de acumulación o almacenamiento”. Durante la historia temprana de la ingeniería de software, de hecho el repositorio era una persona: el programador que debía recordar la ubicación de toda la información relevante para un proyecto de software, quien debía recuperar la información que nunca se escribió y reconstruir la información perdida. Tristemente, usar a una persona como “el centro para acumulación y almacenamiento” (aun conforme con la definición del Webster) no funciona muy bien. En la actualidad, el repositorio es una “cosa”: una base de datos que actúa como el centro de acumulación y de almacenamiento de la información de ingeniería de software. El papel de la persona (el ingeniero del software) es interactuar con el repositorio, usando las herramientas que se integran con él.

### 22.2.1 El papel del repositorio

El repositorio ACS es el conjunto de mecanismos y estructuras de datos que permiten a un equipo de software administrar el cambio en forma efectiva. Proporciona las funciones obvias de un moderno sistema de administración de base de datos, al asegurar integridad, posibilidad de compartir e integración de datos. Además, el repositorio ACS proporciona un centro para la integración de herramientas de software, es fundamental en el flujo del proceso de software y puede reforzar la estructura y el formato uniforme para los productos que son resultado de la ingeniería de software.

Para lograr estas capacidades, el repositorio se define como un metamodelo. El *metamodelo* determina cómo se almacena la información en el repositorio, cómo pueden acceder las herramientas a los datos y cómo pueden verlas los ingenieros de software, cuán bien pueden mantenerse la seguridad y la integridad de los datos y cuán fácilmente puede extenderse el modelo existente para alojar nuevas necesidades.

### 22.2.2 Características y contenido generales

Las características y el contenido del repositorio se entienden mejor al observarlo desde dos perspectivas: qué debe almacenarse en él y qué servicios específicos proporciona. En la figura 22.3 se presenta un desglose detallado de los tipos de representaciones, documentos y otros productos de trabajo.

#### WebRef

En [www.oracle.com/technology/products/repository/index.html](http://www.oracle.com/technology/products/repository/index.html) puede obtenerse un ejemplo de repositorio disponible en el mercado.

Un repositorio robusto proporciona dos clases de servicios diferentes: 1) los mismos tipos de servicios que pueden esperarse de cualquier sistema sofisticado de administración de base de datos y 2) los servicios que son específicos del entorno de ingeniería de software.

Un repositorio que sirve a un equipo de ingeniería de software también debe: 1) integrarse con o directamente apoyar las funciones de administración del proceso, 2) apoyar reglas específicas que gobiernan la función ACS y los datos mantenidos dentro del repositorio, 3) proporcionar una interfaz hacia otras herramientas de ingeniería de software y 4) acomodar almacenamiento de objetos de datos sofisticados (por ejemplo, texto, gráficos, video, audio).

### 22.2.3 Características ACS

Para apoyar el ACS, el repositorio debe tener un conjunto de herramientas que proporcionan apoyo a las siguientes características:

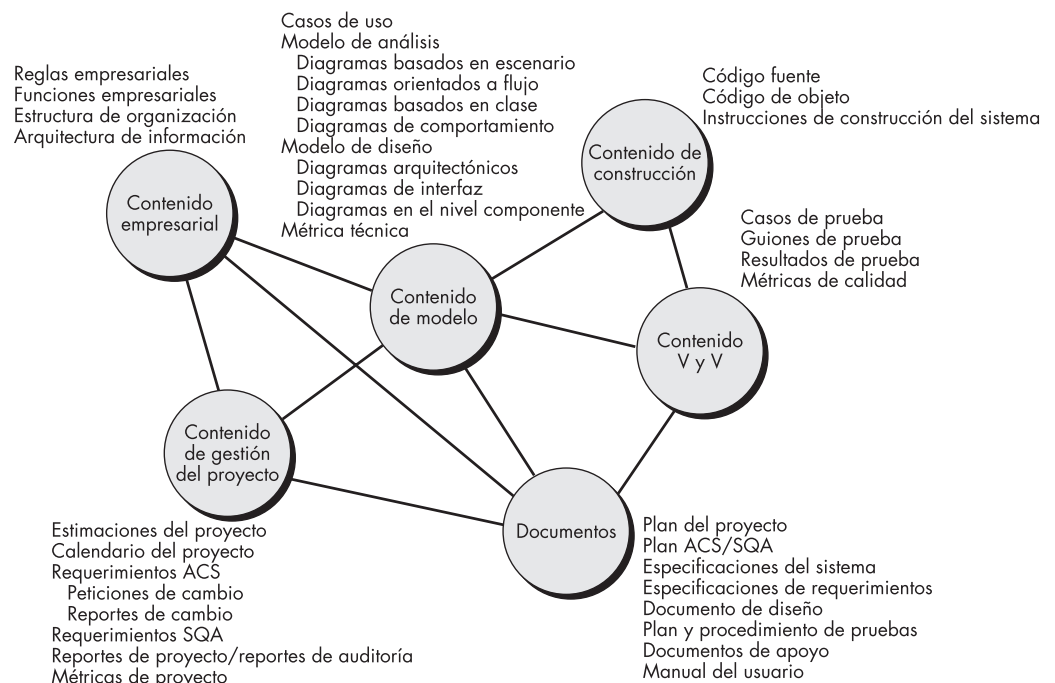
**Versiones.** Conforme avanza el proyecto, se crearán muchas versiones (sección 22.3.2) de productos resultantes individuales. El repositorio debe guardar todas estas versiones para permitir la administración efectiva de los productos liberados y, a los desarrolladores, regresar a versiones anteriores durante las pruebas y la depuración.

#### PUNTO CLAVE

El repositorio debe mantener relacionados los ICS con muchas diferentes versiones del software. Más importante, debe proporcionar los mecanismos para ensamblar dichos ICS en una configuración específica de una versión.

FIGURA 22.3

#### Contenido del repositorio



El repositorio debe controlar una amplia variedad de tipos de objeto, incluidos texto, gráficos, mapas de bits, documentos complejos y objetos únicos, como definiciones de pantalla y reportes, archivos objeto, datos de prueba y resultados. Un repositorio maduro rastrea versiones de objetos con niveles arbitrarios de granularidad; por ejemplo, puede rastrearse una sola definición de datos o un grupo de módulos.

**Rastreo de dependencia y gestión del cambio.** El repositorio administra una amplia variedad de relaciones entre los elementos de datos almacenados en él. En éstos se incluyen relaciones entre entidades y procesos empresariales, entre las partes de un diseño de aplicación, entre componentes de diseño y la arquitectura de información de la empresa, entre elementos de diseño y entregables, etcétera. Algunas de estas relaciones son meras asociaciones, y otras son dependencias o relaciones obligatorias.

La capacidad de seguir la pista de todas estas relaciones es vital para la integridad de la información almacenada en el repositorio y para la generación de entregables con base en él, y es una de las aportaciones más importantes del concepto de repositorio para la mejora del proceso de software. Por ejemplo, si un diagrama de clase UML se modifica, el repositorio puede detectar si clases relacionadas, descripciones de interfaz y componentes de código también requieren modificación y si pueden llevar los ICS afectados a la atención del desarrollador.

**Rastreo de requerimientos.** Esta función especial depende de la administración de vínculos y ofrece la capacidad de rastrear todos los componentes de diseño y construcción, así como entregables que resulten de una especificación de requerimientos determinada (rastreo hacia adelante). Además, proporciona la capacidad de identificar qué requisito genera algún producto de trabajo determinado (rastreo hacia atrás).

**Administración de la configuración.** Una instalación de administración de la configuración sigue la pista a una serie de configuraciones que representa hitos de proyecto específicos o liberaciones de producción.

**Ensayos de auditoría.** Un ensayo de auditoría establece información adicional acerca de cuándo, por qué y quién realiza los cambios. La información acerca de la fuente de los cambios puede ingresarse como atributos de objetos específicos en el repositorio. Un mecanismo de activación de repositorio es útil para que siempre que se modifique un elemento de diseño, se avise al desarrollador, o a la herramienta que se utilice, el inicio de la entrada de la información de auditoría (como la razón para un cambio).

## 22.3 EL PROCESO ACS

### Cita:

"Cualquier cambio, incluso para mejorar, está acompañado de inconvenientes e incomodidades."

Arnold Bennett

El proceso de administración de la configuración del software define una serie de tareas que tienen cuatro objetivos principales: 1) identificar todos los ítems que de manera colectiva definen la configuración del software, 2) administrar los cambios a uno o más de estos ítems, 3) facilitar la construcción de diferentes versiones de una aplicación y 4) garantizar que la calidad del software se conserva conforme la configuración evoluciona con el tiempo.

Un proceso que logra dichos objetivos no necesita ser burocrático o pesado, pero debe caracterizarse de forma que permita a un equipo de software desarrollar respuestas a un conjunto de preguntas complejas:

- ¿Cómo identifica un equipo de software los elementos discretos de una configuración de software?
- ¿Cómo gestiona una organización las muchas versiones existentes de un programa (y su documentación) de manera que permita que el cambio se acomode eficientemente?

**? ¿Qué preguntas debe responder el proceso ACS?**