



## 4

# Ingeniería de requerimientos

### Objetivos

El objetivo de este capítulo es introducir los requerimientos de software y discutir los procesos que hay en el descubrimiento y la documentación de tales requerimientos. Al estudiar este capítulo:

- entenderá los conceptos de requerimientos del usuario y del sistema, así como por qué tales requerimientos se deben escribir en diferentes formas;
- comprenderá las diferencias entre requerimientos de software funcionales y no funcionales;
- reconocerá cómo se organizan los requerimientos dentro de un documento de requerimientos de software;
- conocerá las principales actividades de la ingeniería de requerimientos: adquisición, análisis y validación, así como las relaciones entre dichas actividades;
- analizará por qué es necesaria la administración de requerimientos y cómo ésta apoya otras actividades de la ingeniería de requerimientos.

### Contenido

- 4.1 Requerimientos funcionales y no funcionales
- 4.2 El documento de requerimientos de software
- 4.3 Especificación de requerimientos
- 4.4 Procesos de ingeniería de requerimientos
- 4.5 Adquisición y análisis de requerimientos
- 4.6 Validación de requerimientos
- 4.7 Administración de requerimientos

Los requerimientos para un sistema son descripciones de lo que el sistema debe hacer: el servicio que ofrece y las restricciones en su operación. Tales requerimientos reflejan las necesidades de los clientes por un sistema que atienda cierto propósito, como sería controlar un dispositivo, colocar un pedido o buscar información. Al proceso de descubrir, analizar, documentar y verificar estos servicios y restricciones se le llama ingeniería de requerimientos (IR).

El término “requerimiento” no se usa de manera continua en la industria del software. En algunos casos, un requerimiento es simplemente un enunciado abstracto de alto nivel en un servicio que debe proporcionar un sistema, o bien, una restricción sobre un sistema. En el otro extremo, consiste en una definición detallada y formal de una función del sistema. Davis (1993) explica por qué existen esas diferencias:

*Si una compañía desea otorgar un contrato para un gran proyecto de desarrollo de software, tiene que definir sus necesidades de una forma suficientemente abstracta para que una solución no esté predefinida. Los requerimientos deben redactarse de tal forma que muchos proveedores liciten en pos del contrato, ofreciendo, tal vez, diferentes maneras de cubrir las necesidades de organización del cliente. Una vez otorgado el contrato, el proveedor tiene que escribir con más detalle una definición del sistema para el cliente, de modo que éste comprenda y valide lo que hará el software. Estos documentos suelen nombrarse documentos de requerimientos para el sistema.*

Algunos de los problemas que surgen durante el proceso de ingeniería de requerimientos son resultado del fracaso de hacer una separación clara entre esos diferentes niveles de descripción. En este texto se distinguen con el uso del término “requerimientos del usuario” para representar los requerimientos abstractos de alto nivel; y “requerimientos del sistema” para caracterizar la descripción detallada de lo que el sistema debe hacer. Los requerimientos del usuario y los requerimientos del sistema se definen del siguiente modo:

1. Los requerimientos del usuario son enunciados, en un lenguaje natural junto con diagramas, acerca de qué servicios esperan los usuarios del sistema, y de las restricciones con las cuales éste debe operar.
2. Los requerimientos del sistema son descripciones más detalladas de las funciones, los servicios y las restricciones operacionales del sistema de software. El documento de requerimientos del sistema (llamado en ocasiones especificación funcional) tiene que definir con exactitud lo que se implementará. Puede formar parte del contrato entre el comprador del sistema y los desarrolladores del software.

Los diferentes niveles de requerimientos son útiles debido a que informan sobre el sistema a distintos tipos de lector. La figura 4.1 ilustra la diferencia entre los requerimientos del usuario y del sistema. Este ejemplo de un sistema de administración de pacientes para apoyar la atención a la salud mental (MHC-PMS) muestra cómo los requerimientos del usuario se extienden hacia varios requerimientos del sistema. En la figura 4.1 se observa que el requerimiento del usuario es muy general. Los requerimientos del sistema ofrecen información más específica sobre los servicios y las funciones del sistema que se implementará.

**Figura 4.1**  
Requerimientos  
del usuario  
y requerimientos  
del sistema

#### Definición del requerimiento del usuario

1. El MHC-PMS elaborará mensualmente informes administrativos que revelen el costo de los medicamentos prescritos por cada clínica durante ese mes.

#### Especificación de los requerimientos del sistema

- 1.1 En el último día laboral de cada mes se redactará un resumen de los medicamentos prescritos, su costo y las clínicas que los prescriben.
- 1.2 El sistema elaborará automáticamente el informe que se imprimirá después de las 17:30 del último día laboral del mes.
- 1.3 Se realizará un reporte para cada clínica junto con los nombres de cada medicamento, el número de prescripciones, las dosis prescritas y el costo total de los medicamentos prescritos.
- 1.4 Si los medicamentos están disponibles en diferentes unidades de dosis (por ejemplo, 10 mg, 20 mg) se harán informes por separado para cada unidad de dosis.
- 1.5 El acceso a los informes de costos se restringirá a usuarios autorizados en la lista de control de acceso administrativo.

Es necesario escribir los requerimientos con diferentes niveles de detalle, ya que varios lectores los usarán de distintas formas. La figura 4.2 muestra los posibles lectores de los requerimientos del usuario y los del sistema. De éstos, los primeros por lo general no están interesados en la manera en que se implementará el sistema, y quizá sean administradores a quienes no les atraigan las facilidades detalladas del sistema. Mientras que los segundos necesitan conocer con más precisión qué hará el sistema, ya que están preocupados sobre cómo apoyará los procesos de negocios o porque están inmersos en la implementación del sistema.

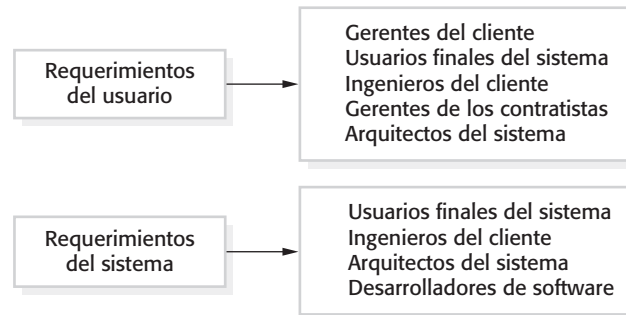
En este capítulo se presenta un panorama “tradicional” de los requerimientos, más que de los requerimientos en los procesos ágiles. Para la mayoría de los sistemas grandes, todavía se presenta una fase de ingeniería de requerimientos claramente identificable, antes de comenzar la implementación del sistema. El resultado es un documento de requerimientos que puede formar parte del contrato de desarrollo del sistema. Desde luego, por lo común hay cambios posteriores a los requerimientos, en tanto que los requerimientos del usuario podrían extenderse como requerimientos de sistema más detallados. Sin embargo, el enfoque ágil para alcanzar, al mismo tiempo, los requerimientos a medida que el sistema se desarrolla rara vez se utiliza en el diseño de sistemas grandes.

## 4.1 Requerimientos funcionales y no funcionales

A menudo, los requerimientos del sistema de software se clasifican como requerimientos funcionales o requerimientos no funcionales:

1. *Requerimientos funcionales* Son enunciados acerca de servicios que el sistema debe proveer, de cómo debería reaccionar el sistema a entradas particulares y de cómo

**Figura 4.2** Lectores de diferentes tipos de especificación de requerimientos



debería comportarse el sistema en situaciones específicas. En algunos casos, los requerimientos funcionales también explican lo que no debe hacer el sistema.

2. *Requerimientos no funcionales* Son limitaciones sobre servicios o funciones que ofrece el sistema. Incluyen restricciones tanto de temporización y del proceso de desarrollo, como impuestas por los estándares. Los requerimientos no funcionales se suelen aplicar al sistema como un todo, más que a características o a servicios individuales del sistema.

En realidad, la distinción entre los diferentes tipos de requerimientos no es tan clara como sugieren estas definiciones sencillas. Un requerimiento de un usuario interesado por la seguridad, como el enunciado que limita el acceso a usuarios autorizados, parecería un requerimiento no funcional. Sin embargo, cuando se desarrolla con más detalle, este requerimiento puede generar otros requerimientos que son evidentemente funcionales, como la necesidad de incluir facilidades de autenticación en el sistema.

Esto muestra que los requerimientos no son independientes y que un requerimiento genera o restringe normalmente otros requerimientos. Por lo tanto, los requerimientos del sistema no sólo detallan los servicios o las características que se requieren del mismo, sino también especifican la funcionalidad necesaria para asegurar que estos servicios y características se entreguen de manera adecuada.

#### 4.1.1 Requerimientos funcionales

Los requerimientos funcionales para un sistema refieren lo que el sistema debe hacer. Tales requerimientos dependen del tipo de software que se esté desarrollando, de los usuarios esperados del software y del enfoque general que adopta la organización cuando se escriben los requerimientos. Al expresarse como requerimientos del usuario, los requerimientos funcionales se describen por lo general de forma abstracta que entiendan los usuarios del sistema. Sin embargo, requerimientos funcionales más específicos del sistema detallan las funciones del sistema, sus entradas y salidas, sus excepciones, etcétera.

Los requerimientos funcionales del sistema varían desde requerimientos generales que cubren lo que tiene que hacer el sistema, hasta requerimientos muy específicos que reflejan maneras locales de trabajar o los sistemas existentes de una organización. Por ejemplo, veamos algunos casos de requerimientos funcionales para el sistema MHC-PMS, que



### Requerimientos de dominio

Los requerimientos de dominio se derivan del dominio de aplicación del sistema, más que a partir de las necesidades específicas de los usuarios del sistema. Pueden ser requerimientos funcionales nuevos por derecho propio, restricciones a los requerimientos funcionales existentes o formas en que deben realizarse cálculos particulares.

El problema con los requerimientos de dominio es que los ingenieros de software no pueden entender las características del dominio en que opera el sistema. Por lo común, no pueden indicar si un requerimiento de dominio se perdió o entró en conflicto con otros requerimientos.

<http://www.SoftwareEngineering-9.com/Web/Requirements/DomainReq.html>

se usan para mantener información de pacientes que reciben tratamiento por problemas de salud mental:

1. Un usuario podrá buscar en todas las clínicas las listas de citas.
2. El sistema elaborará diariamente, para cada clínica, una lista de pacientes que se espera que asistan a cita ese día.
3. Cada miembro del personal que usa el sistema debe identificarse de manera individual con su número de ocho dígitos.

Estos requerimientos funcionales del usuario definen las actividades específicas que debe proporcionar el sistema. Se tomaron del documento de requerimientos del usuario y muestran que los requerimientos funcionales pueden escribirse con diferentes niveles de detalle (contraste los requerimientos 1 y 3).

La inexactitud en la especificación de requerimientos causa muchos problemas en la ingeniería de software. Es natural que un desarrollador de sistemas interprete un requerimiento ambiguo de forma que simplifique su implementación. Sin embargo, con frecuencia, esto no es lo que desea el cliente. Tienen que establecerse nuevos requerimientos y efectuar cambios al sistema. Desde luego, esto aplaza la entrega del sistema y aumenta los costos.

Es el caso del primer ejemplo de requerimiento para el MHC-PMS que establece que un usuario podrá buscar las listas de citas en todas las clínicas. El motivo para este requerimiento es que los pacientes con problemas de salud mental en ocasiones están confundidos. Quizá tengan una cita en una clínica y en realidad acudan a una diferente. De ahí que si tienen una cita, se registrará que asistieron, sin importar la clínica.

Los miembros del personal médico que especifican esto quizás esperen que “buscar” signifique que, dado el nombre de un paciente, el sistema busca dicho nombre en las citas de todas las clínicas. Sin embargo, esto no es claro en el requerimiento. Los desarrolladores del sistema pueden interpretar el requerimiento de forma diferente e implementar una búsqueda, de tal modo que el usuario deba elegir una clínica y luego realizar la búsqueda. Evidentemente, esto implicará más entradas del usuario y tomará más tiempo.

En principio, la especificación de los requerimientos funcionales de un sistema debe ser completa y consistente. Totalidad significa que deben definirse todos los servicios requeridos por el usuario. Consistencia quiere decir que los requerimientos tienen que evitar definiciones contradictorias. En la práctica, para sistemas complejos grandes, es

casi imposible lograr la consistencia y la totalidad de los requerimientos. Una causa para ello es la facilidad con que se cometen errores y omisiones al escribir especificaciones para sistemas complejos. Otra es que hay muchos participantes en un sistema grande. Un participante es un individuo o una función que se ve afectado de alguna forma por el sistema. Los participantes tienen diferentes necesidades, pero con frecuencia son inconsistentes. Tales inconsistencias tal vez no sean evidentes cuando se especifican por primera vez los requerimientos, de modo que en la especificación se incluyen requerimientos inconsistentes. Los problemas suelen surgir sólo después de un análisis en profundidad o después de que se entregó el sistema al cliente.

#### 4.1.2 Requerimientos no funcionales

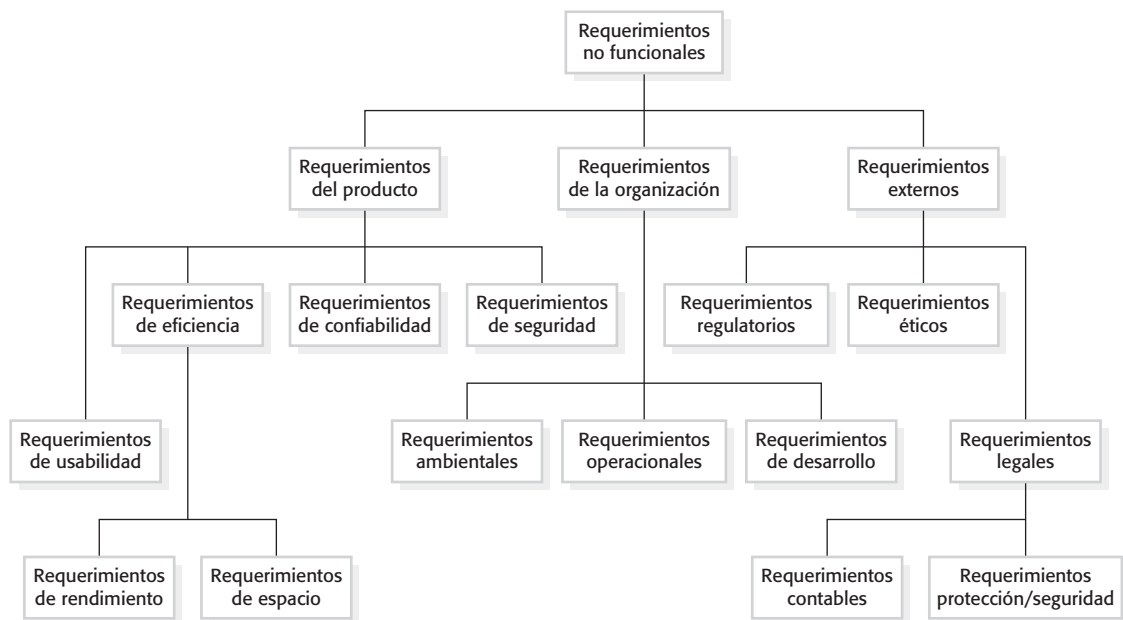
Los requerimientos no funcionales, como indica su nombre, son requerimientos que no se relacionan directamente con los servicios específicos que el sistema entrega a sus usuarios. Pueden relacionarse con propiedades emergentes del sistema, como fiabilidad, tiempo de respuesta y uso de almacenamiento. De forma alternativa, pueden definir restricciones sobre la implementación del sistema, como las capacidades de los dispositivos I/O o las representaciones de datos usados en las interfaces con otros sistemas.

Los requerimientos no funcionales, como el rendimiento, la seguridad o la disponibilidad, especifican o restringen por lo general características del sistema como un todo. Los requerimientos no funcionales a menudo son más significativos que los requerimientos funcionales individuales. Es común que los usuarios del sistema encuentren formas para trabajar en torno a una función del sistema que realmente no cubre sus necesidades. No obstante, el fracaso para cubrir los requerimientos no funcionales haría que todo el sistema fuera inútil. Por ejemplo, si un sistema de aeronave no cubre sus requerimientos de fiabilidad, no será certificado para su operación como dispositivo seguro; si un sistema de control embebido fracasa para cubrir sus requerimientos de rendimiento, no operarán correctamente las funciones de control.

Aunque es posible identificar con regularidad cuáles componentes de sistema implementan requerimientos funcionales específicos (por ejemplo, hay componentes de formato que implementan requerimientos de informe), por lo general es más difícil relacionar componentes con requerimientos no funcionales. La implementación de dichos requerimientos puede propagarse a lo largo del sistema. Para esto existen dos razones:

1. Los requerimientos no funcionales afectan más la arquitectura global de un sistema que los componentes individuales. Por ejemplo, para garantizar que se cumplan los requerimientos de rendimiento, quizá se deba organizar el sistema para minimizar las comunicaciones entre componentes.
2. Un requerimiento no funcional individual, como un requerimiento de seguridad, podría generar algunos requerimientos funcionales relacionados que definan nuevos servicios del sistema que se requieran. Además, también podría generar requerimientos que restrinjan los requerimientos ya existentes.

Los requerimientos no funcionales surgen a través de necesidades del usuario, debido a restricciones presupuestales, políticas de la organización, necesidad de interoperabilidad con otro software o sistemas de hardware, o factores externos como regulaciones de seguridad o legislación sobre privacidad. La figura 4.3 es una clasificación de requerimientos



**Figura 4.3** Tipos de requerimientos no funcionales

no funcionales. Observe a partir de este diagrama que los requerimientos no funcionales provienen de características requeridas del software (requerimientos del producto), la organización que desarrolla el software (requerimientos de la organización) o de fuentes externas:

1. *Requerimientos del producto* Estos requerimientos especifican o restringen el comportamiento del software. Los ejemplos incluyen requerimientos de rendimiento sobre qué tan rápido se debe ejecutar el sistema y cuánta memoria requiere, requerimientos de fiabilidad que establecen la tasa aceptable de fallas, requerimientos de seguridad y requerimientos de usabilidad.
2. *Requerimientos de la organización* Son requerimientos de sistemas amplios, derivados de políticas y procedimientos en la organización del cliente y del desarrollador. Los ejemplos incluyen requerimientos del proceso operacional que definen cómo se usará el sistema, requerimientos del proceso de desarrollo que especifican el lenguaje de programación, estándares del entorno o el proceso de desarrollo a utilizar, y requerimientos ambientales que definen el entorno de operación del sistema.
3. *Requerimientos externos* Este término cubre todos los requerimientos derivados de factores externos al sistema y su proceso de desarrollo. En ellos se incluyen requerimientos regulatorios que establecen lo que debe hacer el sistema para ser aprobado en su uso por un regulador, como sería un banco central; requerimientos legislativos que tienen que seguirse para garantizar que el sistema opere conforme a la ley, y requerimientos éticos que garanticen que el sistema será aceptable para sus usuarios y el público en general.

La figura 4.4 muestra ejemplos de requerimientos del producto, de la organización y requerimientos externos tomados del MHC-PMS, cuyos requerimientos de usuario se



**REQUERIMIENTO DEL PRODUCTO**

El MHC-PMS estará disponible en todas las clínicas durante las horas de trabajo normales (lunes a viernes, de 8:30 a 17:30). En cualquier día, los tiempos muertos dentro de las horas laborales normales no rebasarán los cinco segundos.

**REQUERIMIENTOS DE LA ORGANIZACIÓN**

Los usuarios del sistema MHC-PMS se acreditarán a sí mismos con el uso de la tarjeta de identidad de la autoridad sanitaria.

**REQUERIMIENTOS EXTERNOS**

Como establece la HStan-03-2006-priv, el sistema implementará provisiones para la privacidad del paciente.

**Figura 4.4** Ejemplos de requerimientos no funcionales en el MHC-PMS

introdujeron en la sección 4.1.1. El requerimiento del producto es un requerimiento de disponibilidad que define cuándo estará disponible el sistema y el tiempo muerto permitido cada día. No dice algo sobre la funcionalidad del MHC-PMS e identifica con claridad una restricción que deben considerar los diseñadores del sistema.

El requerimiento de la organización especifica cómo se autentican los usuarios en el sistema. La autoridad sanitaria que opera el sistema se mueve hacia un procedimiento de autenticación estándar para cualquier software donde, en vez de que los usuarios tengan un nombre de conexión (login), pasan su tarjeta de identidad por un lector para identificarse a sí mismos. El requerimiento externo se deriva de la necesidad de que el sistema esté conforme con la legislación de privacidad. Evidentemente, la privacidad es un asunto muy importante en los sistemas de atención a la salud, y el requerimiento especifica que el sistema debe desarrollarse conforme a un estándar de privacidad nacional.

Un problema común con requerimientos no funcionales es que los usuarios o clientes con frecuencia proponen estos requerimientos como metas generales, como facilidad de uso, capacidad de que el sistema se recupere de fallas, o rapidez de respuesta al usuario. Las metas establecen buenas intenciones; no obstante, ocasionan problemas a los desarrolladores del sistema, pues dejan espacio para la interpretación y la disputa posterior una vez que se entregue el sistema. Por ejemplo, la siguiente meta del sistema es típica de cómo un administrador expresa los requerimientos de usabilidad:

*Para el personal médico debe ser fácil usar el sistema, y este último debe organizarse de tal forma que minimice los errores del usuario.*

Lo anterior se escribió para mostrar cómo podría expresarse la meta como un requerimiento no funcional “comprobable”. Aun cuando es imposible comprobar de manera objetiva la meta del sistema, en la siguiente descripción se puede incluir, al menos, la instrumentación de software para contar los errores cometidos por los usuarios cuando prueban el sistema.

*Después de cuatro horas de capacitación, el personal médico usará todas las funciones del sistema. Después de esta capacitación, los usuarios experimentados no deberán superar el promedio de dos errores cometidos por hora de uso del sistema.*

Siempre que sea posible, se deberán escribir de manera cuantitativa los requerimientos no funcionales, de manera que puedan ponerse objetivamente a prueba. La figura 4.5 muestra las métricas que se utilizan para especificar propiedades no funcionales del sistema.



Propiedad	Medida
Rapidez	Transacciones/segundo procesadas Tiempo de respuesta usuario/evento Tiempo de regeneración de pantalla
Tamaño	Mbytes Número de chips ROM
Facilidad de uso	Tiempo de capacitación Número de cuadros de ayuda
Fiabilidad	Tiempo medio para falla Probabilidad de indisponibilidad Tasa de ocurrencia de falla Disponibilidad
Robustez	Tiempo de reinicio después de falla Porcentaje de eventos que causan falla Probabilidad de corrupción de datos en falla
Portabilidad	Porcentaje de enunciados dependientes de objetivo Número de sistemas objetivo

**Figura 4.5** Métricas para especificar requerimientos no funcionales

Usted puede medir dichas características cuando el sistema se pone a prueba para comprobar si éste cumple o no cumple con sus requerimientos no funcionales.

En la práctica, los usuarios de un sistema suelen encontrar difícil traducir sus metas en requerimientos mensurables. Para algunas metas, como la mantenibilidad, no hay métricas para usarse. En otros casos, incluso cuando sea posible la especificación cuantitativa, los clientes no logran relacionar sus necesidades con dichas especificaciones. No comprenden qué significa algún número que define la fiabilidad requerida (por así decirlo), en términos de su experiencia cotidiana con los sistemas de cómputo. Más aún, el costo por verificar objetivamente los requerimientos no funcionales mensurables suele ser muy elevado, y los clientes que pagan por el sistema quizá piensen que dichos costos no están justificados.

Los requerimientos no funcionales entran a menudo en conflicto e interactúan con otros requerimientos funcionales o no funcionales. Por ejemplo, el requerimiento de autenticación en la figura 4.4 requiere, indiscutiblemente, la instalación de un lector de tarjetas en cada computadora unida al sistema. Sin embargo, podría haber otro requerimiento que solicite acceso móvil al sistema desde las computadoras portátiles de médicos o enfermeras. Por lo general, las computadoras portátiles no están equipadas con lectores de tarjeta, de modo que, ante tales circunstancias, probablemente deba permitirse algún método de autenticación alternativo.

En la práctica, en el documento de requerimientos, resulta difícil separar los requerimientos funcionales de los no funcionales. Si los requerimientos no funcionales se expresan por separado de los requerimientos funcionales, las relaciones entre ambos serían difíciles de entender. No obstante, se deben destacar de manera explícita los requerimientos que están claramente relacionados con las propiedades emergentes del sistema, como el rendimiento o la fiabilidad. Esto se logra al ponerlos en una sección separada del documento de requerimientos o al distinguirlos, en alguna forma, de otros requerimientos del sistema.



### Estándares del documento de requerimientos

Algunas organizaciones grandes, como el Departamento de Defensa estadounidense y el Institute of Electrical and Electronic Engineers (IEEE), definieron estándares para los documentos de requerimientos. Comúnmente son muy genéricos, pero útiles como base para desarrollar estándares organizativos más detallados. El IEEE es uno de los proveedores de estándares mejor conocidos y desarrolló un estándar para la estructura de documentos de requerimientos. Este estándar es más adecuado para sistemas como comando militar y sistemas de control que tienen un largo tiempo de vida y, por lo general, los diseña un grupo de organizaciones.

<http://www.SoftwareEngineering-9.com/Web/Requirements/IEEE-standard.html>

Los requerimientos no funcionales, como los requerimientos de fiabilidad, protección y confidencialidad, son en particular importantes para los sistemas fundamentales. En el capítulo 12 se incluyen estos requerimientos, donde se describen técnicas específicas para definir requerimientos de confiabilidad y seguridad.

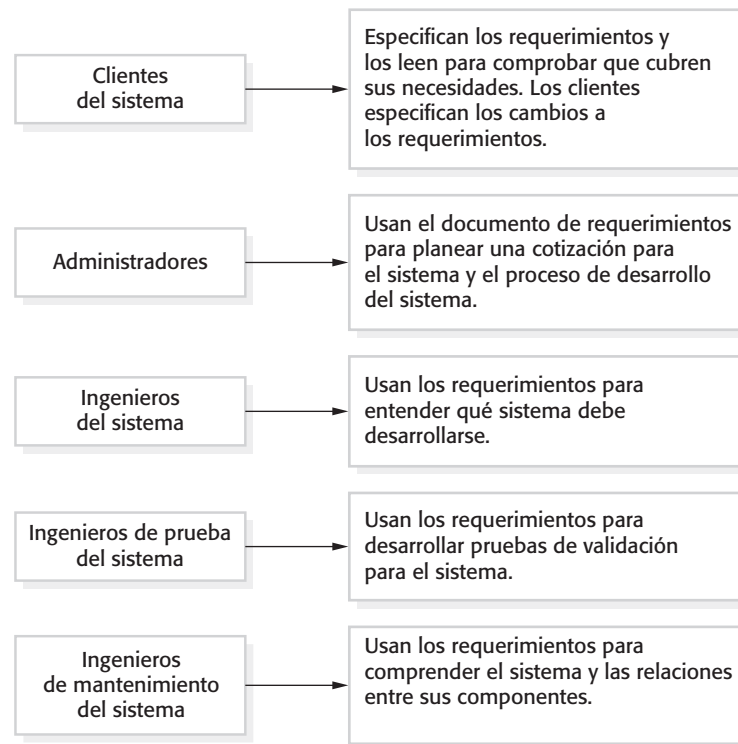
## 4.2 El documento de requerimientos de software

El documento de requerimientos de software (llamado algunas veces especificación de requerimientos de software o SRS) es un comunicado oficial de lo que deben implementar los desarrolladores del sistema. Incluye tanto los requerimientos del usuario para un sistema, como una especificación detallada de los requerimientos del sistema. En ocasiones, los requerimientos del usuario y del sistema se integran en una sola descripción. En otros casos, los requerimientos del usuario se definen en una introducción a la especificación de requerimientos del sistema. Si hay un gran número de requerimientos, los requerimientos del sistema detallados podrían presentarse en un documento aparte.

Son esenciales los documentos de requerimientos cuando un contratista externo diseña el sistema de software. Sin embargo, los métodos de desarrollo ágiles argumentan que los requerimientos cambian tan rápidamente que un documento de requerimientos se vuelve obsoleto tan pronto como se escribe, así que el esfuerzo se desperdicia en gran medida. En lugar de un documento formal, los enfoques como la programación extrema (Beck, 1999) recopilan de manera incremental requerimientos del usuario y los escriben en tarjetas como historias de usuario. De esa manera, el usuario da prioridad a los requerimientos para su implementación en el siguiente incremento del sistema.

Este enfoque es adecuado para sistemas empresariales donde los requerimientos son inestables. Sin embargo, aún resulta útil escribir un breve documento de apoyo que defina los requerimientos de la empresa y los requerimientos de confiabilidad para el sistema; es fácil olvidar los requerimientos que se aplican al sistema como un todo, cuando uno se enfoca en los requerimientos funcionales para la siguiente liberación del sistema.

El documento de requerimientos tiene un conjunto variado de usuarios, desde el administrador ejecutivo de la organización que paga por el sistema, hasta los ingenieros responsables del desarrollo del software. La figura 4.6, tomada del libro del autor con Gerald Kotonya sobre ingeniería de requerimientos (Kotonya y Sommerville, 1998), muestra a los posibles usuarios del documento y cómo ellos lo utilizan.



**Figura 4.6** Usuarios de un documento de requerimientos

La diversidad de posibles usuarios significa que el documento de requerimientos debe ser un compromiso entre la comunicación de los requerimientos a los clientes, la definición de los requerimientos con detalle preciso para desarrolladores y examinadores, y la inclusión de información sobre la posible evolución del sistema. La información de cambios anticipados ayuda tanto a los diseñadores del sistema a evitar decisiones de diseño restrictivas, como a los ingenieros de mantenimiento del sistema que deben adaptar el sistema a los nuevos requerimientos.

El nivel de detalle que se incluya en un documento de requerimientos depende del tipo de sistema a diseñar y el proceso de desarrollo utilizado. Los sistemas críticos necesitan tener requerimientos detallados porque la seguridad y la protección también deben analizarse de forma pormenorizada. Cuando el sistema lo desarrolla una compañía independiente (por ejemplo, mediante la subcontratación), deben detallarse y precisarse las especificaciones del sistema. Si se utiliza un proceso de desarrollo iterativo interno, entonces el documento de requerimientos suele ser mucho menos detallado y cualquier ambigüedad puede resolverse durante el desarrollo del sistema.

La figura 4.7 indica una posible organización para un documento de requerimientos basada en un estándar del IEEE para documentos de requerimientos (IEEE, 1998). Este estándar es genérico y se adapta a usos específicos. En este caso, el estándar se extendió para incluir información de la evolución prevista del sistema. Esta información ayuda a los encargados del sistema y permite a los diseñadores incluir soporte para características futuras del sistema.

Naturalmente, la información que se incluya en un documento de requerimientos depende del tipo de software que se va a desarrollar y del enfoque para el desarrollo que se use. Si se adopta un enfoque evolutivo para un producto de software (por ejemplo), el

Capítulo	Descripción
Prefacio	Debe definir el número esperado de lectores del documento, así como describir su historia de versiones, incluidas las causas para la creación de una nueva versión y un resumen de los cambios realizados en cada versión.
Introducción	Describe la necesidad para el sistema. Debe detallar brevemente las funciones del sistema y explicar cómo funcionará con otros sistemas. También tiene que indicar cómo se ajusta el sistema en los objetivos empresariales o estratégicos globales de la organización que comisiona el software.
Glosario	Define los términos técnicos usados en el documento. No debe hacer conjeturas sobre la experiencia o la habilidad del lector.
Definición de requerimientos del usuario	Aquí se representan los servicios que ofrecen al usuario. También, en esta sección se describen los requerimientos no funcionales del sistema. Esta descripción puede usar lenguaje natural, diagramas u otras observaciones que sean comprensibles para los clientes. Deben especificarse los estándares de producto y proceso que tienen que seguirse.
Arquitectura del sistema	Este capítulo presenta un panorama de alto nivel de la arquitectura anticipada del sistema, que muestra la distribución de funciones a través de los módulos del sistema. Hay que destacar los componentes arquitectónicos que sean de reutilización.
Especificación de requerimientos del sistema	Debe representar los requerimientos funcionales y no funcionales con más detalle. Si es preciso, también pueden detallarse más los requerimientos no funcionales. Pueden definirse las interfaces a otros sistemas.
Modelos del sistema	Pueden incluir modelos gráficos del sistema que muestren las relaciones entre componentes del sistema, el sistema y su entorno. Ejemplos de posibles modelos son los modelos de objeto, modelos de flujo de datos o modelos de datos semánticos.
Evolución del sistema	Describe los supuestos fundamentales sobre los que se basa el sistema, y cualquier cambio anticipado debido a evolución de hardware, cambio en las necesidades del usuario, etc. Esta sección es útil para los diseñadores del sistema, pues los ayuda a evitar decisiones de diseño que restringirían probablemente futuros cambios al sistema.
Apéndices	Brindan información específica y detallada que se relaciona con la aplicación a desarrollar; por ejemplo, descripciones de hardware y bases de datos. Los requerimientos de hardware definen las configuraciones, mínima y óptima, del sistema. Los requerimientos de base de datos delimitan la organización lógica de los datos usados por el sistema y las relaciones entre datos.
Índice	Pueden incluirse en el documento varios índices. Así como un índice alfabético normal, uno de diagramas, un índice de funciones, etcétera.

**Figura 4.7** Estructura de un documento de requerimientos

documento de requerimientos dejará fuera muchos de los capítulos detallados que se sugirieron anteriormente. El enfoque estará en especificar los requerimientos del usuario y los requerimientos no funcionales de alto nivel del sistema. En este caso, diseñadores y programadores usan su criterio para decidir cómo cubrir los requerimientos establecidos del usuario para el sistema.

Sin embargo, cuando el software sea parte de un proyecto de sistema grande que incluya la interacción de sistemas de hardware y software, será necesario por lo general

**Problemas con el uso de lenguaje natural para la especificación de requerimientos**

La flexibilidad del lenguaje natural, que es tan útil para la especificación, causa problemas frecuentemente. Hay espacio para escribir requerimientos poco claros, y los lectores (los diseñadores) pueden malinterpretar los requerimientos porque tienen un antecedente diferente al del usuario. Es fácil mezclar muchos requerimientos en una sola oración y quizá sea difícil estructurar los requerimientos en lenguaje natural.

<http://www.SoftwareEngineering-9.com/Web/Requirements/NL-problems.html>

definir los requerimientos a un nivel detallado. Esto significa que es probable que los documentos de requerimientos sean muy largos y deban incluir la mayoría, si no es que todos, los capítulos que se muestran en la figura 4.7. Para documentos extensos, es muy importante incluir una tabla de contenido global y un índice del documento, de manera que los lectores encuentren con facilidad la información que necesitan.

### 4.3 Especificación de requerimientos

La especificación de requerimientos es el proceso de escribir, en un documento de requerimientos, los requerimientos del usuario y del sistema. De manera ideal, los requerimientos del usuario y del sistema deben ser claros, sin ambigüedades, fáciles de entender, completos y consistentes. Esto en la práctica es difícil de lograr, pues los participantes interpretan los requerimientos de formas diferentes y con frecuencia en los requerimientos hay conflictos e inconsistencias inherentes.

Los requerimientos del usuario para un sistema deben describir los requerimientos funcionales y no funcionales, de forma que sean comprensibles para los usuarios del sistema que no cuentan con un conocimiento técnico detallado. De manera ideal, deberían especificar sólo el comportamiento externo del sistema. El documento de requerimientos no debe incluir detalles de la arquitectura o el diseño del sistema. En consecuencia, si usted escribe los requerimientos del usuario, no tiene que usar jerga de software, anotaciones estructuradas o formales. Debe escribir los requerimientos del usuario en lenguaje natural, con tablas y formas sencillas, así como diagramas intuitivos.

Los requerimientos del sistema son versiones extendidas de los requerimientos del usuario que los ingenieros de software usan como punto de partida para el diseño del sistema. Añaden detalles y explican cómo el sistema debe brindar los requerimientos del usuario. Se pueden usar como parte del contrato para la implementación del sistema y, por lo tanto, deben ser una especificación completa y detallada de todo el sistema.

Idealmente, los requerimientos del sistema deben describir de manera simple el comportamiento externo del sistema y sus restricciones operacionales. No tienen que ocuparse de cómo se diseña o implementa el sistema. Sin embargo, al nivel de detalle requerido para especificar por completo un sistema de software complejo, es prácticamente imposible excluir toda la información de diseño. Para ello existen varias razones:

1. Tal vez se tenga que diseñar una arquitectura inicial del sistema para ayudar a estructurar la especificación de requerimientos. Los requerimientos del sistema se organizan

Notación	Descripción
Enunciados en lenguaje natural	Los requerimientos se escriben al usar enunciados numerados en lenguaje natural. Cada enunciado debe expresar un requerimiento.
Lenguaje natural estructurado	Los requerimientos se escriben en lenguaje natural en una forma o plantilla estándar. Cada campo ofrece información de un aspecto del requerimiento.
Lenguajes de descripción de diseño	Este enfoque usa un lenguaje como un lenguaje de programación, pero con características más abstractas para especificar los requerimientos al definir un modelo operacional del sistema. Aunque en la actualidad este enfoque se usa raras veces, aún tiene utilidad para especificaciones de interfaz.
Anotaciones gráficas	Los modelos gráficos, complementados con anotaciones de texto, sirven para definir los requerimientos funcionales del sistema; los casos de uso del UML y los diagramas de secuencia se emplean de forma común.
Especificaciones matemáticas	Dichas anotaciones se basan en conceptos matemáticos como máquinas o conjuntos de estado finito. Aunque tales especificaciones sin ambigüedades pueden reducir la imprecisión en un documento de requerimientos, la mayoría de los clientes no comprenden una especificación formal. No pueden comprobar que representa lo que quieren y por ello tienen reticencia para aceptarlo como un contrato de sistema.

**Figura 4.8** Formas de escribir una especificación de requerimientos del sistema

de acuerdo con los diferentes subsistemas que constituyen el sistema. Como veremos en los capítulos 6 y 18, esta definición arquitectónica es esencial si usted quiere reutilizar componentes de software al implementar el sistema.

2. En la mayoría de los casos, los sistemas deben interoperar con los sistemas existentes, lo cual restringe el diseño e impone requerimientos sobre el nuevo sistema.
3. Quizá sea necesario el uso de una arquitectura específica para cubrir los requerimientos no funcionales (como la programación N-versión para lograr fiabilidad, que se estudia en el capítulo 13). Un regulador externo, que precise certificar que dicho sistema es seguro, puede especificar que se utilice un diseño arquitectónico ya avalado.

Los requerimientos del usuario se escriben casi siempre en lenguaje natural, complementado con diagramas y tablas adecuados en el documento de requerimientos. Los requerimientos del sistema se escriben también en lenguaje natural, pero de igual modo se utilizan otras notaciones basadas en formas, modelos gráficos del sistema o modelos matemáticos del sistema. La figura 4.8 resume las posibles anotaciones que podrían usarse para escribir requerimientos del sistema.

Los modelos gráficos son más útiles cuando es necesario mostrar cómo cambia un estado o al describir una secuencia de acciones. Los gráficos de secuencia UML y los gráficos de estado, que se explican en el capítulo 5, exponen la secuencia de acciones que ocurren en respuesta a cierto mensaje o evento. En ocasiones, se usan especificaciones matemáticas formales con la finalidad de describir los requerimientos para sistemas de protección o seguridad críticos, aunque rara vez se usan en otras circunstancias. Este enfoque para escribir especificaciones se explica en el capítulo 12.

3.2 Si se requiere, cada 10 minutos el sistema medirá el azúcar en la sangre y administrará insulina. *(Los cambios de azúcar en la sangre son relativamente lentos, de manera que no son necesarias mediciones más frecuentes; la medición menos periódica podría conducir a niveles de azúcar innecesariamente elevados.)*

3.6 Cada minuto, el sistema debe correr una rutina de autoevaluación, con las condiciones a probar y las acciones asociadas definidas en la tabla 1. *(Una rutina de autoevaluación puede detectar problemas de hardware y software, y prevenir al usuario sobre el hecho de que la operación normal puede ser imposible.)*

**Figura 4.9**  
Ejemplo de  
requerimientos  
para el sistema  
de software de  
la bomba de  
insulina

### 4.3.1 Especificación en lenguaje natural

Desde los albores de la ingeniería de software, el lenguaje natural se usa para escribir los requerimientos de software. Es expresivo, intuitivo y universal. También es potencialmente vago, ambiguo y su significado depende de los antecedentes del lector. Como resultado, hay muchas propuestas para formas alternativas de escribir los requerimientos. Sin embargo, ninguna se ha adoptado de manera amplia, por lo que el lenguaje natural seguirá siendo la forma más usada para especificar los requerimientos del sistema y del software.

Para minimizar la interpretación errónea al escribir los requerimientos en lenguaje natural, se recomienda seguir algunos lineamientos sencillos:

1. Elabore un formato estándar y asegúrese de que todas las definiciones de requerimientos se adhieran a dicho formato. Al estandarizar el formato es menos probable cometer omisiones y más sencillo comprobar los requerimientos. El formato que usa el autor expresa el requerimiento en una sola oración. A cada requerimiento de usuario se asocia un enunciado de razones para explicar por qué se propuso el requerimiento. Las razones también pueden incluir información sobre quién planteó el requerimiento (la fuente del requerimiento), de modo que usted conozca a quién consultar en caso de que cambie el requerimiento.
2. Utilice el lenguaje de manera clara para distinguir entre requerimientos obligatorios y deseables. Los primeros son requerimientos que el sistema debe soportar y, por lo general, se escriben en futuro “debe ser”. En tanto que los requerimientos deseables no son necesarios y se escriben en tiempo pospretérito o como condicional “debería ser”.
3. Use texto resaltado (negrilla, cursiva o color) para seleccionar partes clave del requerimiento.
4. No deduzca que los lectores entienden el lenguaje técnico de la ingeniería de software. Es fácil que se malinterpreten palabras como “arquitectura” y “módulo”. Por lo tanto, debe evitar el uso de jerga, abreviaturas y acrónimos.
5. Siempre que sea posible, asocie una razón con cada requerimiento de usuario. La razón debe explicar por qué se incluyó el requerimiento. Es particularmente útil cuando los requerimientos cambian, pues ayuda a decidir cuáles cambios serían indeseables.

La figura 4.9 ilustra cómo se usan dichos lineamientos. Incluye dos requerimientos para el software embebido para la bomba de insulina automatizada, que se introdujo en el capítulo 1. Usted puede descargar la especificación completa de los requerimientos de la bomba de insulina en las páginas Web del libro.



**Bomba de insulina/Software de control/SRS/3.3.2**

<b>Función</b>	Calcula dosis de insulina: nivel seguro de azúcar.
<b>Descripción</b>	Calcula la dosis de insulina que se va a suministrar cuando la medición del nivel de azúcar actual esté en zona segura entre 3 y 7 unidades.
<b>Entradas</b>	Lectura del azúcar actual (r2), las dos lecturas previas (r0 y r1).
<b>Fuente</b>	Lectura del azúcar actual del sensor. Otras lecturas de la memoria.
<b>Salidas</b>	CompDose: la dosis de insulina a administrar.
<b>Destino</b>	Ciclo de control principal.
<b>Acción</b>	CompDose es cero si es estable el nivel de azúcar, o cae o si aumenta el nivel pero disminuye la tasa de aumento. Si el nivel se eleva y la tasa de aumento crece, CompDose se calcula entonces al dividir la diferencia entre el nivel de azúcar actual y el nivel previo entre 4 y redondear el resultado. Si la suma se redondea a cero, en tal caso CompDose se establece en la dosis mínima que puede entregarse.
<b>Requerimientos</b>	Dos lecturas previas, de modo que puede calcularse la tasa de cambio del nivel de azúcar.
<b>Precondición</b>	El depósito de insulina contiene al menos la dosis individual de insulina máxima permitida.
<b>Postcondición</b>	r0 se sustituye con r1, luego r1 se sustituye con r2.
<b>Efectos colaterales</b>	Ninguno.

**Figura 4.10 4.3.2 Especificaciones estructuradas**

Especificación estructurada de un requerimiento para una bomba de insulina

El lenguaje natural estructurado es una manera de escribir requerimientos del sistema, donde está limitada la libertad del escritor de requerimientos y todos éstos se anotan en una forma estándar. Aunque este enfoque conserva la mayoría de la expresividad y comprensibilidad del lenguaje natural, asegura que haya cierta uniformidad sobre la especificación. Las anotaciones en lenguaje estructurado emplean plantillas para especificar requerimientos del sistema. La especificación utiliza constructos de lenguaje de programación para mostrar alternativas e iteración, y destaca elementos clave con el uso de sombreado o de fuentes distintas.

Los Robertson (Robertson y Robertson, 1999), en su libro del método de ingeniería de requerimientos VOLERE, recomiendan que se escriban los requerimientos del usuario inicialmente en tarjetas, un requerimiento por tarjeta. Proponen algunos campos en cada tarjeta, tales como razones de los requerimientos, dependencias en otros requerimientos, fuente de los requerimientos, materiales de apoyo, etcétera. Lo anterior es similar al enfoque utilizado en el ejemplo de la especificación estructurada que se muestra en la figura 4.10.

Para usar un enfoque estructurado que especifique los requerimientos de sistema, hay que definir una o más plantillas estándar para requerimientos, y representar dichas plantillas como formas estructuradas. La especificación puede estructurarse sobre los objetos manipulados por el sistema, las funciones que el sistema realiza o los eventos procesados por el sistema. En la figura 4.10 se muestra un ejemplo de una especificación basada en la forma, en este caso, una que define cómo calcular la dosis de insulina a administrar cuando el azúcar en la sangre está dentro de una banda segura.

Condición	Acción
Nivel de azúcar en descenso ( $r2 < r1$ )	CompDose = 0
Nivel de azúcar estable ( $r2 = r1$ )	CompDose = 0
Nivel de azúcar creciente y tasa de incremento decreciente ( $(r2 - r1) < (r1 - r0)$ )	CompDose = 0
Nivel de azúcar creciente y tasa de incremento estable o creciente ( $(r2 - r1) \geq (r1 - r0)$ )	CompDose = round $((r2 - r1)/4)$ If resultado redondeado = 0 then CompDose = MinimumDose

**Figura 4.11**  
Especificación tabular  
del cálculo para una  
bomba de insulina

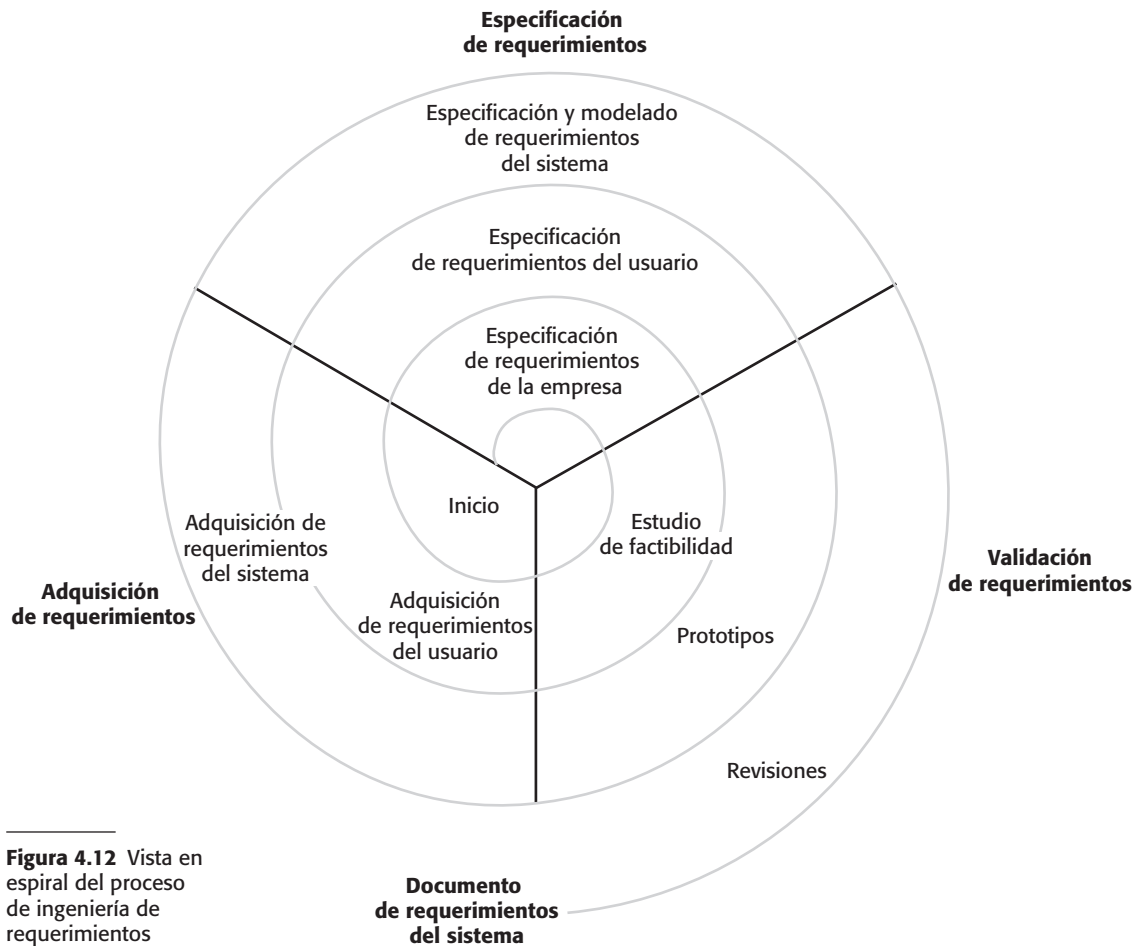
Cuando use una forma estándar para especificar requerimientos funcionales, debe incluir la siguiente información:

1. Una descripción de la función o entidad a especificar.
2. Una descripción de sus entradas y sus procedencias.
3. Una descripción de sus salidas y a dónde se dirigen.
4. Información sobre los datos requeridos para el cálculo u otras entidades en el sistema que se utilizan (la parte “requiere”).
5. Una descripción de la acción que se va a tomar.
6. Si se usa un enfoque funcional, una precondition establece lo que debe ser verdadero antes de llamar a la función, y una postcondición especifica lo que es verdadero después de llamar a la función.
7. Una descripción de los efectos colaterales (si acaso hay alguno) de la operación.

Al usar especificaciones estructuradas se eliminan algunos de los problemas de la especificación en lenguaje natural. La variabilidad en la especificación se reduce y los requerimientos se organizan de forma más efectiva. Sin embargo, en ocasiones todavía es difícil escribir requerimientos sin ambigüedades, en particular cuando deben especificarse cálculos complejos (por ejemplo, cómo calcular la dosis de insulina).

Para enfrentar este problema se puede agregar información extra a los requerimientos en lenguaje natural, por ejemplo, con el uso de tablas o modelos gráficos del sistema. Éstos pueden mostrar cómo proceden los cálculos, cambia el estado del sistema, interactúan los usuarios con el sistema y se realizan las secuencias de acciones.

Las tablas son particularmente útiles cuando hay algunas posibles situaciones alternas y se necesita describir las acciones a tomar en cada una de ellas. La bomba de insulina fundamenta sus cálculos del requerimiento de insulina, en la tasa de cambio de los niveles de azúcar en la sangre. Las tasas de cambio se calculan con las lecturas, actual y anterior. La figura 4.11 es una descripción tabular de cómo se usa la tasa de cambio del azúcar en la sangre, para calcular la cantidad de insulina por suministrar.



**Figura 4.12** Vista en espiral del proceso de ingeniería de requerimientos

## 4.4 Procesos de ingeniería de requerimientos

Como vimos en el capítulo 2, los procesos de ingeniería de requerimientos incluyen cuatro actividades de alto nivel. Éstas se enfocan en valorar si el sistema es útil para la empresa (estudio de factibilidad), descubrir requerimientos (adquisición y análisis), convertir dichos requerimientos en alguna forma estándar (especificación) y comprobar que los requerimientos definan realmente el sistema que quiere el cliente (validación). En la figura 2.6 se mostró esto como proceso secuencial; sin embargo, en la práctica, la ingeniería de requerimientos es un proceso iterativo donde las actividades están entrelazadas.

La figura 4.12 presenta este entrelazamiento. Las actividades están organizadas como un proceso iterativo alrededor de una espiral, y la salida es un documento de requerimientos del sistema. La cantidad de tiempo y esfuerzo dedicados a cada actividad en cada iteración depende de la etapa del proceso global y el tipo de sistema que está siendo desarrollado. En el inicio del proceso, se empleará más esfuerzo para comprender los requerimientos empresariales de alto nivel y los no funcionales, así como los requerimientos del



### Estudios de factibilidad

Un estudio de factibilidad es un breve estudio enfocado que debe realizarse con oportunidad en el proceso de IR. Debe responder tres preguntas clave: a) ¿El sistema contribuye con los objetivos globales de la organización? b) ¿El sistema puede implementarse dentro de la fecha y el presupuesto usando la tecnología actual? c) ¿El sistema puede integrarse con otros sistemas que se utilicen?

Si la respuesta a cualquiera de estas preguntas es negativa, probablemente no sea conveniente continuar con el proyecto.

<http://www.SoftwareEngineering-9.com/Web/Requirements/FeasibilityStudy.html>

usuario para el sistema. Más adelante en el proceso, en los anillos exteriores de la espiral, se dedicará más esfuerzo a la adquisición y comprensión de los requerimientos detallados del sistema.

Este modelo en espiral acomoda enfoques al desarrollo, donde los requerimientos se elaboraron con diferentes niveles de detalle. El número de iteraciones de la espiral tiende a variar, de modo que la espiral terminará después de adquirir algunos o todos los requerimientos del usuario. Se puede usar el desarrollo ágil en vez de la creación de prototipos, de manera que se diseñen en conjunto los requerimientos y la implementación del sistema.

Algunas personas consideran la ingeniería de requerimientos como el proceso de aplicar un método de análisis estructurado, tal como el análisis orientado a objetos (Larman, 2002). Esto implica analizar el sistema y desarrollar un conjunto de modelos gráficos del sistema, como los modelos de caso de uso, que luego sirven como especificación del sistema. El conjunto de modelos describe el comportamiento del sistema y se anota con información adicional que describe, por ejemplo, el rendimiento o la fiabilidad requeridos del sistema.

Aunque los métodos estructurados desempeñan un papel en el proceso de ingeniería de requerimientos, hay mucho más ingeniería de requerimientos de la que se cubre con dichos métodos. La adquisición de requerimientos, en particular, es una actividad centrada en la gente, y a las personas no les gustan las restricciones impuestas por modelos de sistema rígidos.

Prácticamente en todos los sistemas cambian los requerimientos. Las personas implicadas desarrollan una mejor comprensión de qué quieren que haga el software; la organización que compra el sistema cambia; se hacen modificaciones al hardware, al software y al entorno organizacional del sistema. Al proceso de administrar tales requerimientos cambiantes se le llama administración de requerimientos, tema que se trata en la sección 4.7.

## 4.5 Adquisición y análisis de requerimientos

Después de un estudio de factibilidad inicial, la siguiente etapa del proceso de ingeniería de requerimientos es la adquisición y el análisis de requerimientos. En esta actividad, los ingenieros de software trabajan con clientes y usuarios finales del sistema para descubrir el dominio de aplicación, qué servicios debe proporcionar el sistema, el desempeño requerido de éste, las restricciones de hardware, etcétera.



**Figura 4.13** El proceso de adquisición y análisis de requerimientos

En una organización, la adquisición y el análisis de requerimientos pueden involucrar a diversas clases de personas. Un participante en el sistema es quien debe tener alguna influencia directa o indirecta sobre los requerimientos del mismo. Los participantes incluyen a usuarios finales que interactuarán con el sistema, y a cualquiera en una organización que resultará afectada por él. Otros participantes del sistema pueden ser los ingenieros que desarrollan o mantienen otros sistemas relacionados, administradores de negocios, expertos de dominio y representantes de asociaciones sindicales.

En la figura 4.13 se muestra un modelo del proceso de adquisición y análisis. Cada organización tendrá su versión o ejemplificación de este modelo general, dependiendo de factores locales, tales como experiencia del personal, tipo de sistema a desarrollar, estándares usados, etcétera.

Las actividades del proceso son:

1. *Descubrimiento de requerimientos* Éste es el proceso de interactuar con los participantes del sistema para descubrir sus requerimientos. También los requerimientos de dominio de los participantes y la documentación se descubren durante esta actividad. Existen numerosas técnicas complementarias que pueden usarse para el descubrimiento de requerimientos, las cuales se estudian más adelante en esta sección.
2. *Clasificación y organización de requerimientos* Esta actividad toma la compilación no estructurada de requerimientos, agrupa requerimientos relacionados y los organiza en grupos coherentes. La forma más común de agrupar requerimientos es usar un modelo de la arquitectura del sistema, para identificar subsistemas y asociar los requerimientos con cada subsistema. En la práctica, la ingeniería de requerimientos y el diseño arquitectónico no son actividades separadas completamente.
3. *Priorización y negociación de requerimientos* Inevitablemente, cuando intervienen diversos participantes, los requerimientos entrarán en conflicto. Esta actividad se preocupa por priorizar los requerimientos, así como por encontrar y resolver conflictos de requerimientos mediante la negociación. Por lo general, los participantes tienen que reunirse para resolver las diferencias y estar de acuerdo con el compromiso de los requerimientos.

4. *Especificación de requerimientos* Los requerimientos se documentan e ingresan en la siguiente ronda de la espiral. Pueden producirse documentos de requerimientos formales o informales, como se estudia en la sección 4.3.

La figura 4.13 muestra que la adquisición y el análisis de requerimientos es un proceso iterativo con retroalimentación continua de cada actividad a otras actividades. El ciclo del proceso comienza con el descubrimiento de requerimientos y termina con la documentación de los requerimientos. La comprensión de los requerimientos por parte del analista mejora con cada ronda del ciclo. El ciclo concluye cuando está completo el documento de requerimientos.

La adquisición y la comprensión de los requerimientos por parte de los participantes del sistema es un proceso difícil por diferentes razones:

1. Los participantes con frecuencia no saben lo que quieren de un sistema de cómputo, excepto en términos muy generales; pueden encontrar difícil articular qué quieren que haga el sistema; pueden hacer peticiones inalcanzables porque no saben qué es factible y qué no lo es.
2. Los participantes en un sistema expresan naturalmente los requerimientos con sus términos y conocimientos implícitos de su trabajo. Los ingenieros de requerimientos, sin experiencia en el dominio del cliente, podrían no entender dichos requerimientos.
3. Diferentes participantes tienen distintos requerimientos y pueden expresarlos en variadas formas. Los ingenieros de requerimientos deben descubrir todas las fuentes potenciales de requerimientos e identificar similitudes y conflictos.
4. Factores políticos llegan a influir en los requerimientos de un sistema. Los administradores pueden solicitar requerimientos específicos del sistema, porque éstos les permitirán aumentar su influencia en la organización.
5. El ambiente económico y empresarial donde ocurre el análisis es dinámico. Inevitablemente cambia durante el proceso de análisis. Puede cambiar la importancia de requerimientos particulares; o bien, tal vez surjan nuevos requerimientos de nuevos participantes a quienes no se consultó originalmente.

Resulta ineludible que diferentes participantes tengan diversas visiones de la importancia y prioridad de los requerimientos y, algunas veces, dichas visiones están en conflicto. Durante el proceso, usted deberá organizar negociaciones regulares con los participantes, de forma que se alcancen compromisos. Es imposible complacer por completo a cada participante, pero, si algunos suponen que sus visiones no se consideraron de forma adecuada, quizás intenten deliberadamente socavar el proceso de IR.

En la etapa de especificación de requerimientos, los requerimientos adquiridos hasta el momento se documentan de tal forma que puedan usarse para ayudar al hallazgo de requerimientos. En esta etapa, podría generarse una primera versión del documento de requerimientos del sistema, con secciones faltantes y requerimientos incompletos. De modo alternativo, los requerimientos pueden documentarse en una forma completamente diferente (por ejemplo, en una hoja de cálculo o en tarjetas). Escribir requerimientos en tarjetas suele ser muy efectivo, ya que los participantes las administran, cambian y organizan con facilidad.



### Puntos de vista

Un punto de vista es una forma de recopilar y organizar un conjunto de requerimientos de un grupo de participantes que cuentan con algo en común. Por lo tanto, cada punto de vista incluye una serie de requerimientos del sistema. Los puntos de vista pueden provenir de usuarios finales, administradores, etcétera. Ayudan a identificar a los individuos que brindan información sobre sus requerimientos y a estructurar los requerimientos para análisis.

<http://www.SoftwareEngineering-9.com/Web/Requirements/Viewpoints.html>

## 4.5.1 Descubrimiento de requerimientos

El descubrimiento de requerimientos (llamado a veces adquisición de requerimientos) es el proceso de recopilar información sobre el sistema requerido y los sistemas existentes, así como de separar, a partir de esta información, los requerimientos del usuario y del sistema. Las fuentes de información durante la fase de descubrimiento de requerimientos incluyen documentación, participantes del sistema y especificaciones de sistemas similares. La interacción con los participantes es a través de entrevistas y observaciones, y pueden usarse escenarios y prototipos para ayudar a los participantes a entender cómo será el sistema.

Los participantes varían desde administradores y usuarios finales de un sistema hasta participantes externos como los reguladores, quienes certifican la aceptabilidad del sistema. Por ejemplo, los participantes que se incluyen para el sistema de información de pacientes en atención a la salud mental son:

1. Pacientes cuya información se registra en el sistema.
2. Médicos que son responsables de valorar y tratar a los pacientes.
3. Enfermeros que coordinan, junto con los médicos, las consultas y suministran algunos tratamientos.
4. Recepcionistas que administran las citas médicas de los pacientes.
5. Personal de TI que es responsable de instalar y mantener el sistema.
6. Un director de ética médica que debe garantizar que el sistema cumpla con los lineamientos éticos actuales de la atención al paciente.
7. Encargados de atención a la salud que obtienen información administrativa del sistema.
8. Personal de archivo médico que es responsable de garantizar que la información del sistema se conserve, y se implementen de manera adecuada los procedimientos de mantenimiento del archivo.

Además de los participantes del sistema, se observa que los requerimientos también pueden venir del dominio de aplicación y de otros sistemas que interactúan con el sistema a especificar. Todos ellos deben considerarse durante el proceso de adquisición de requerimientos.

Todas estas diferentes fuentes de requerimientos (participantes, dominio, sistemas) se representan como puntos de vista del sistema, y cada visión muestra un subconjunto de los



requerimientos para el sistema. Diferentes puntos de vista de un problema enfocan el problema de diferentes formas. Sin embargo, sus perspectivas no son totalmente independientes, sino que por lo general se traslapan, de manera que tienen requerimientos comunes. Usted puede usar estos puntos de vista para estructurar tanto el descubrimiento como la documentación de los requerimientos del sistema.

### 4.5.2 Entrevistas

---

Las entrevistas formales o informales con participantes del sistema son una parte de la mayoría de los procesos de ingeniería de requerimientos. En estas entrevistas, el equipo de ingeniería de requerimientos formula preguntas a los participantes sobre el sistema que actualmente usan y el sistema que se va a desarrollar. Los requerimientos se derivan de las respuestas a dichas preguntas. Las entrevistas son de dos tipos:

1. Entrevistas cerradas, donde los participantes responden a un conjunto de preguntas preestablecidas.
2. Entrevistas abiertas, en las cuales no hay agenda predefinida. El equipo de ingeniería de requerimientos explora un rango de conflictos con los participantes del sistema y, como resultado, desarrolla una mejor comprensión de sus necesidades.

En la práctica, las entrevistas con los participantes son por lo general una combinación de ambas. Quizá se deba obtener la respuesta a ciertas preguntas, pero eso a menudo conduce a otros temas que se discuten en una forma menos estructurada. Rara vez funcionan bien las discusiones completamente abiertas. Con frecuencia debe plantear algunas preguntas para comenzar y mantener la entrevista enfocada en el sistema que se va a desarrollar.

Las entrevistas son valiosas para lograr una comprensión global sobre qué hacen los participantes, cómo pueden interactuar con el nuevo sistema y las dificultades que enfrentan con los sistemas actuales. A las personas les gusta hablar acerca de sus trabajos, así que por lo general están muy dispuestas a participar en entrevistas. Sin embargo, las entrevistas no son tan útiles para comprender los requerimientos desde el dominio de la aplicación.

Por dos razones resulta difícil asimilar el conocimiento de dominio a través de entrevistas:

1. Todos los especialistas en la aplicación usan terminología y jerga que son específicos de un dominio. Es imposible que ellos discutan los requerimientos de dominio sin usar este tipo de lenguaje. Por lo general, usan la terminología en una forma precisa y sutil, que para los ingenieros de requerimientos es fácil de malinterpretar.
2. Cierta conocimiento del dominio es tan familiar a los participantes que encuentran difícil de explicarlo, o bien, creen que es tan fundamental que no vale la pena mencionarlo. Por ejemplo, para un bibliotecario no es necesario decir que todas las adquisiciones deben catalogarse antes de agregarlas al acervo. Sin embargo, esto quizá no sea obvio para el entrevistador y, por lo tanto, es posible que no lo tome en cuenta en los requerimientos.

Las entrevistas tampoco son una técnica efectiva para adquirir conocimiento sobre los requerimientos y las restricciones de la organización, porque existen relaciones sutiles de poder entre los diferentes miembros en la organización. Las estructuras publicadas de

la organización rara vez coinciden con la realidad de la toma de decisiones en una organización, pero los entrevistados quizá no deseen revelar a un extraño la estructura real, sino la teórica. En general, la mayoría de las personas se muestran renuentes a discutir los conflictos políticos y organizacionales que afecten los requerimientos.

Los entrevistadores efectivos poseen dos características:

1. Tienen mentalidad abierta, evitan ideas preconcebidas sobre los requerimientos y escuchan a los participantes. Si el participante aparece con requerimientos sorprendentes, entonces tienen disposición para cambiar su mentalidad acerca del sistema.
2. Instan al entrevistado con una pregunta de trampolín para continuar la plática, dar una propuesta de requerimientos o trabajar juntos en un sistema de prototipo. Cuando se pregunta al individuo “dime qué quieres” es improbable que alguien consiga información útil. Encuentran mucho más sencillo hablar en un contexto definido que en términos generales.

La información de las entrevistas se complementa con otra información del sistema de documentación que describe los procesos empresariales o los sistemas existentes, las observaciones del usuario, etcétera. En ocasiones, además de los documentos del sistema, la información de la entrevista puede ser la única fuente de datos sobre los requerimientos del sistema. Sin embargo, la entrevista por sí misma está expuesta a perder información esencial y, por consiguiente, debe usarse junto con otras técnicas de adquisición de requerimientos.

### 4.5.3 Escenarios

Por lo general, las personas encuentran más sencillo vincularse con ejemplos reales que con descripciones abstractas. Pueden comprender y criticar un escenario sobre cómo interactuar con un sistema de software. Los ingenieros de requerimientos usan la información obtenida de esta discusión para formular los verdaderos requerimientos del sistema.

Los escenarios son particularmente útiles para detallar un bosquejo de descripción de requerimientos. Se trata de ejemplos sobre descripciones de sesiones de interacción. Cada escenario abarca comúnmente una interacción o un número pequeño de interacciones posibles. Se desarrollan diferentes formas de escenarios y se ofrecen varios tipos de información con diversos niveles de detalle acerca del sistema. Las historias que se usan en programación extrema, estudiadas en el capítulo 3, son un tipo de escenario de requerimientos.

Un escenario comienza con un bosquejo de la interacción. Durante el proceso de adquisición, se suman detalles a éste para crear una representación completa de dicha interacción. En su forma más general, un escenario puede incluir:

1. Una descripción de qué esperan el sistema y los usuarios cuando inicia el escenario.
2. Una descripción en el escenario del flujo normal de los eventos.
3. Una descripción de qué puede salir mal y cómo se manejaría.
4. Información de otras actividades que estén en marcha al mismo tiempo.
5. Una descripción del estado del sistema cuando termina el escenario.

**SUPOSICIÓN INICIAL:**

El paciente observa a un auxiliar médico que elabora un registro en el sistema y recaba información personal de aquél (nombre, dirección, edad, etcétera). Una enfermera ingresa en el sistema y obtiene la historia médica.

**NORMAL:**

La enfermera busca al paciente por su nombre completo. Si hay más de un paciente con el mismo apellido, para identificarlo se usa el nombre y la fecha de nacimiento.

La enfermera elige la opción de menú y añade la historia médica.

Inmediatamente la enfermera sigue una serie de indicadores (*prompt*) del sistema para ingresar información de consultas en otras instituciones, sobre problemas de salud mental (entrada libre de texto), condiciones médicas existentes (la enfermera selecciona las condiciones del menú), medicamentos administrados actualmente (seleccionados del menú), alergias (texto libre) y vida familiar (formato).

**QUÉ PUEDE SALIR MAL:**

Si no existe el registro del paciente o no puede encontrarse, la enfermera debe crear un nuevo registro e ingresar información personal.

Las condiciones o los medicamentos del paciente no se ingresan en el menú. La enfermera debe elegir la opción "otro" e ingresar texto libre que describa la condición/medicamento.

El paciente no puede/no proporciona información acerca de su historia médica. La enfermera tiene que ingresar a texto libre que registre la incapacidad/renuencia a brindar información. El sistema debe imprimir el formato de exclusión estándar que menciona que la falta de información podría significar que el tratamiento esté limitado o demorado. Esto tiene que firmarlo el paciente.

**OTRAS ACTIVIDADES:**

Mientras se ingresa la información, otros miembros del personal pueden consultar los registros, pero no editarlos.

**ESTADO DEL SISTEMA A COMPLETAR:**

Ingreso del usuario. El registro del paciente, incluida su historia médica, se integra en la base de datos, se agrega un registro a la bitácora (log) del sistema que indica el tiempo de inicio y terminación de la sesión y la enfermera a cargo.

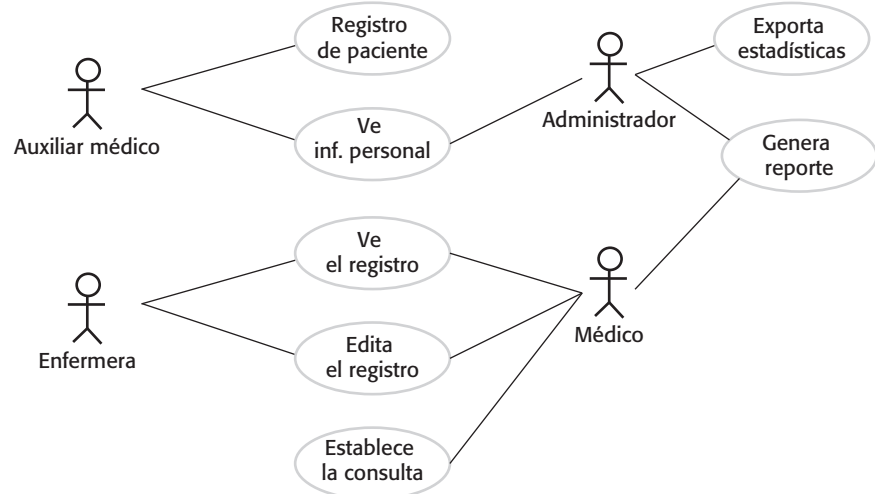
**Figura 4.14** Escenario para recabar historia médica en MHC-PMS

La adquisición basada en escenario implica trabajar con los participantes para identificar escenarios y captar detalles a incluir en dichos escenarios. Estos últimos pueden escribirse como texto, complementarse con diagramas, tomas de pantallas, etcétera. De forma alternativa, es posible usar un enfoque más estructurado, como los escenarios de evento o casos de uso.

Como ejemplo de un simple escenario de texto, considere cómo usaría el MHC-PMS para ingresar datos de un nuevo paciente (figura 4.14). Cuando un nuevo paciente asiste a una clínica, un auxiliar médico crea un nuevo registro y agrega información personal (nombre, edad, etcétera). Después, una enfermera entrevista al paciente y recaba su historia médica. Luego, el paciente tiene una consulta inicial con un médico que lo diagnostica y, si es adecuado, recomienda un tratamiento. El escenario muestra lo que sucede cuando se recaba la historia médica.

#### 4.5.4 Casos de uso

Los casos de uso son una técnica de descubrimiento de requerimientos que se introdujo por primera vez en el método Objectory (Jacobson *et al.*, 1993). Ahora se ha convertido



**Figura 4.15** Casos de uso para el MHC-PMS

en una característica fundamental del modelado de lenguaje unificado. En su forma más sencilla, un caso de uso identifica a los actores implicados en una interacción, y nombra el tipo de interacción. Entonces, esto se complementa con información adicional que describe la interacción con el sistema. La información adicional puede ser una descripción textual, o bien, uno o más modelos gráficos como una secuencia UML o un gráfico de estado.

Los casos de uso se documentan con el empleo de un diagrama de caso de uso de alto nivel. El conjunto de casos de uso representa todas las interacciones posibles que se describirán en los requerimientos del sistema. Los actores en el proceso, que pueden ser individuos u otros sistemas, se representan como figuras sencillas. Cada clase de interacción se constituye como una elipse con etiqueta. Líneas vinculan a los actores con la interacción. De manera opcional, se agregan puntas de flecha a las líneas para mostrar cómo se inicia la interacción. Esto se ilustra en la figura 4.15, que presenta algunos de los casos de uso para el sistema de información del paciente.

No hay distinción tajante y rápida entre escenarios y casos de uso. Algunas personas consideran que cada caso de uso es un solo escenario; otras, como sugieren Stevens y Pooley (2006), encapsulan un conjunto de escenarios en un solo caso de uso. Cada escenario es un solo hilo a través del caso de uso. Por lo tanto, habría un escenario para la interacción normal, más escenarios para cada posible excepción. En la práctica, es posible usarlos en cualquier forma.

Los casos de uso identifican las interacciones individuales entre el sistema y sus usuarios u otros sistemas. Cada caso de uso debe documentarse con una descripción textual. Entonces pueden vincularse con otros modelos en el UML que desarrollará el escenario con más detalle. Por ejemplo, una breve descripción del caso de uso *Establece la consulta* de la figura 4.15 sería:

*El establecimiento de consulta permite que dos o más médicos, que trabajan en diferentes consultorios, vean el mismo registro simultáneamente. Un médico inicia la consulta al elegir al individuo involucrado de un menú desplegable de médicos que estén en línea. Entonces el registro del paciente se despliega en sus pantallas,*

*pero sólo el médico que inicia puede editar el registro. Además, se crea una ventana de chat de texto para ayudar a coordinar las acciones. Se supone que, de manera separada, se establecerá una conferencia telefónica para comunicación por voz.*

Los escenarios y los casos de uso son técnicas efectivas para adquirir requerimientos de los participantes que interactúan directamente con el sistema. Cada tipo de interacción puede representarse como caso de uso. Sin embargo, debido a que se enfocan en interacciones con el sistema, no son tan efectivas para adquirir restricciones o requerimientos empresariales y no funcionales de alto nivel, ni para descubrir requerimientos de dominio.

El UML es un estándar *de facto* para modelado orientado a objetos, así que los casos de uso y la adquisición basada en casos ahora se utilizan ampliamente para adquisición de requerimientos. Los casos de uso se estudian en el capítulo 5, y se muestra cómo se emplean junto con otros modelos del sistema para documentar el diseño de un sistema.

#### 4.5.5 Etnografía

---

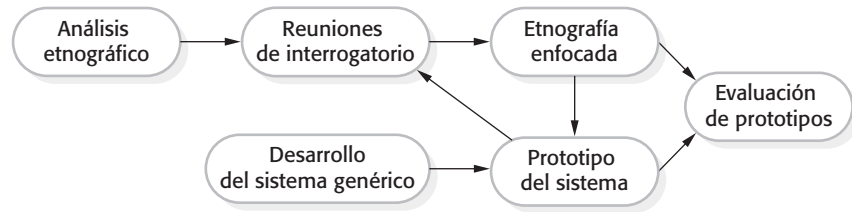
Los sistemas de software no existen aislados. Se usan en un contexto social y organizacional, y dicho escenario podría derivar o restringir los requerimientos del sistema de software. A menudo satisfacer dichos requerimientos sociales y organizacionales es crítico para el éxito del sistema. Una razón por la que muchos sistemas de software se entregan, y nunca se utilizan, es que sus requerimientos no consideran de manera adecuada cómo afectaría el contexto social y organizacional la operación práctica del sistema.

La etnografía es una técnica de observación que se usa para entender los procesos operacionales y ayudar a derivar requerimientos de apoyo para dichos procesos. Un analista se adentra en el ambiente laboral donde se usará el sistema. Observa el trabajo diario y toma notas acerca de las tareas existentes en que intervienen los participantes. El valor de la etnografía es que ayuda a descubrir requerimientos implícitos del sistema que reflejan las formas actuales en que trabaja la gente, en vez de los procesos formales definidos por la organización.

Las personas con frecuencia encuentran muy difícil articular los detalles de su trabajo, porque es una segunda forma de vida para ellas. Entienden su trabajo, pero tal vez no su relación con otras funciones en la organización. Los factores sociales y organizacionales que afectan el trabajo, que no son evidentes para los individuos, sólo se vuelven claros cuando los percibe un observador sin prejuicios. Por ejemplo, un grupo de trabajo puede organizarse de modo que sus miembros conozcan el trabajo de los demás y se suplan entre sí cuando alguien se ausenta. Es probable que esto no se mencione durante una entrevista, pues el grupo podría no verlo como una parte integral de su función.

Suchman (1987) fue una de las primeras en usar la etnografía para estudiar el trabajo en la oficina. Ella descubrió que las prácticas reales del trabajo son más ricas, más complejas y más dinámicas que los modelos simples supuestos por los sistemas de automatización administrativa. La diferencia entre el trabajo supuesto y el real fue la razón más importante por la que dichos sistemas de oficina no tenían un efecto significativo sobre la productividad. Crabtree (2003) analiza desde entonces una amplia gama de estudios, y describe, en

**Figura 4.16** Etnografía y creación de prototipos para análisis de requerimientos



general, el uso de la etnografía en el diseño de sistemas. El autor ha investigado métodos para integrar la etnografía en el proceso de ingeniería de software, mediante su vinculación con los métodos de la ingeniería de requerimientos (Viller y Sommerville, 1999; Viller y Sommerville, 2000) y patrones para documentar la interacción en sistemas cooperativos (Martin *et al.*, 2001; Martin *et al.*, 2002; Martin y Sommerville, 2004).

La etnografía es muy efectiva para descubrir dos tipos de requerimientos:

1. Requerimientos que se derivan de la forma en que realmente trabaja la gente, en vez de la forma en la cual las definiciones del proceso indican que debería trabajar. Por ejemplo, los controladores de tráfico aéreo pueden desactivar un sistema de alerta de conflicto que detecte una aeronave con trayectoria de vuelo que se cruza, aun cuando los procedimientos de control normales especifiquen que es obligatorio usar tal sistema. Ellos deliberadamente dejan a la aeronave sobre la ruta de conflicto durante breves momentos, para ayudarse a dirigir el espacio aéreo. Su estrategia de control está diseñada para garantizar que dichas aeronaves se desvíen antes de que haya problemas, y consideran que la alarma de alerta de conflicto los distrae de su trabajo.
2. Requerimientos que se derivan de la cooperación y el conocimiento de las actividades de otras personas. Por ejemplo, los controladores de tráfico aéreo pueden usar el conocimiento del trabajo de otros controladores para predecir el número de aeronaves que entrarán a su sector de control. Entonces, modifican sus estrategias de control dependiendo de dicha carga de trabajo prevista. Por lo tanto, un sistema ATC automatizado debería permitir a los controladores en un sector tener cierta visibilidad del trabajo en sectores adyacentes.

La etnografía puede combinarse con la creación de prototipos (figura 4.16). La etnografía informa del desarrollo del prototipo, de modo que se requieren menos ciclos de refinamiento del prototipo. Más aún, la creación de prototipos se enfoca en la etnografía al identificar problemas y preguntas que entonces pueden discutirse con el etnógrafo. Siendo así, éste debe buscar las respuestas a dichas preguntas durante la siguiente fase de estudio del sistema (Sommerville *et al.*, 1993).

Los estudios etnográficos pueden revelar detalles críticos de procesos, que con frecuencia se pierden con otras técnicas de adquisición de requerimientos. Sin embargo, debido a su enfoque en el usuario final, no siempre es adecuado para descubrir requerimientos de la organización o de dominio. No en todos los casos se identifican nuevas características que deben agregarse a un sistema. En consecuencia, la etnografía no es un enfoque completo para la adquisición por sí misma, y debe usarse para complementar otros enfoques, como el análisis de casos de uso.



### Revisiones de requerimientos

Una revisión de requerimientos es un proceso donde un grupo de personas del cliente del sistema y el desarrollador del sistema leen con detalle el documento de requerimientos y buscan errores, anomalías e inconsistencias. Una vez detectados y registrados, recae en el cliente y el desarrollador la labor de negociar cómo resolver los problemas identificados.

<http://www.SoftwareEngineering-9.com/Web/Requirements/Reviews.html>

## 4.6 Validación de requerimientos

La validación de requerimientos es el proceso de verificar que los requerimientos definan realmente el sistema que en verdad quiere el cliente. Se traslapa con el análisis, ya que se interesa por encontrar problemas con los requerimientos. La validación de requerimientos es importante porque los errores en un documento de requerimientos pueden conducir a grandes costos por tener que rehacer, cuando dichos problemas se descubren durante el desarrollo del sistema o después de que éste se halla en servicio.

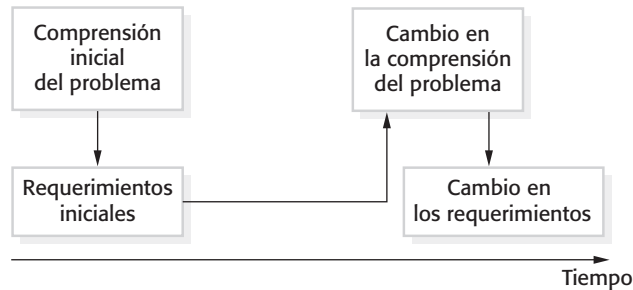
En general, el costo por corregir un problema de requerimientos al hacer un cambio en el sistema es mucho mayor que reparar los errores de diseño o codificación. La razón es que un cambio a los requerimientos significa generalmente que también deben cambiar el diseño y la implementación del sistema. Más aún, el sistema debe entonces ponerse a prueba de nuevo.

Durante el proceso de validación de requerimientos, tienen que realizarse diferentes tipos de comprobaciones sobre los requerimientos contenidos en el documento de requerimientos. Dichas comprobaciones incluyen:

1. *Comprobaciones de validez* Un usuario quizá crea que necesita un sistema para realizar ciertas funciones. Sin embargo, con mayor consideración y análisis se logra identificar las funciones adicionales o diferentes que se requieran. Los sistemas tienen diversos participantes con diferentes necesidades, y cualquier conjunto de requerimientos es inevitablemente un compromiso a través de la comunidad de participantes.
2. *Comprobaciones de consistencia* Los requerimientos en el documento no deben estar en conflicto. Esto es, no debe haber restricciones contradictorias o descripciones diferentes de la misma función del sistema.
3. *Comprobaciones de totalidad* El documento de requerimientos debe incluir requerimientos que definan todas las funciones y las restricciones pretendidas por el usuario del sistema.
4. *Comprobaciones de realismo* Al usar el conocimiento de la tecnología existente, los requerimientos deben comprobarse para garantizar que en realidad pueden implementarse. Dichas comprobaciones también tienen que considerar el presupuesto y la fecha para el desarrollo del sistema.
5. *Verificabilidad* Para reducir el potencial de disputas entre cliente y contratista, los requerimientos del sistema deben escribirse siempre de manera que sean verificables. Esto significa que usted debe ser capaz de escribir un conjunto de pruebas que demuestren que el sistema entregado cumpla cada requerimiento especificado.



**Figura 4.17**  
Evolución de los  
requerimientos



Hay algunas técnicas de validación de requerimientos que se usan individualmente o en conjunto con otras:

1. *Revisiones de requerimientos* Los requerimientos se analizan sistemáticamente usando un equipo de revisores que verifican errores e inconsistencias.
2. *Creación de prototipos* En esta aproximación a la validación, se muestra un modelo ejecutable del sistema en cuestión a los usuarios finales y clientes. Así, ellos podrán experimentar con este modelo para constatar si cubre sus necesidades reales.
3. *Generación de casos de prueba* Los requerimientos deben ser comprobables. Si las pruebas para los requerimientos se diseñan como parte del proceso de validación, esto revela con frecuencia problemas en los requerimientos. Si una prueba es difícil o imposible de diseñar, esto generalmente significa que los requerimientos serán difíciles de implementar, por lo que deberían reconsiderarse. El desarrollo de pruebas a partir de los requerimientos del usuario antes de escribir cualquier código es una pieza integral de la programación extrema.

No hay que subestimar los problemas incluidos en la validación de requerimientos. A final de cuentas, es difícil demostrar que un conjunto de requerimientos, de hecho, no cubre las necesidades de los usuarios. Estos últimos necesitan una imagen del sistema en operación, así como comprender la forma en que dicho sistema se ajustará a su trabajo. Es difícil, incluso para profesionales de la computación experimentados, realizar este tipo de análisis abstracto, y más aún para los usuarios del sistema. Como resultado, rara vez usted encontrará todos los problemas de requerimientos durante el proceso de validación de requerimientos. Es inevitable que haya más cambios en los requerimientos para corregir omisiones y malas interpretaciones, después de acordar el documento de requerimientos.

## 4.7 Administración de requerimientos

Los requerimientos para los grandes sistemas de software siempre cambian. Una razón es que dichos sistemas se desarrollaron por lo general para resolver problemas “horrorosos”: aquellos problemas que no se pueden definir por completo. Como el problema no se logra definir por completo, los requerimientos del software están condenados también a estar incompletos. Durante el proceso de software, la comprensión que los participantes tienen de los problemas cambia constantemente (figura 4.17). Entonces, los requerimientos del sistema también deben evolucionar para reflejar esa visión cambiante del problema.



### Requerimientos duraderos y volátiles

Algunos requerimientos son más susceptibles a cambiar que otros. Los requerimientos duraderos son los requerimientos que se asocian con las actividades centrales, de lento cambio, de una organización. También estos requerimientos se relacionan con actividades laborales fundamentales. Por el contrario, los requerimientos volátiles tienen más probabilidad de cambio. Se asocian por lo general con actividades de apoyo que reflejan cómo la organización hace su trabajo más que el trabajo en sí.

<http://www.SoftwareEngineering-9.com/Web/Requirements/EnduringReq.html>

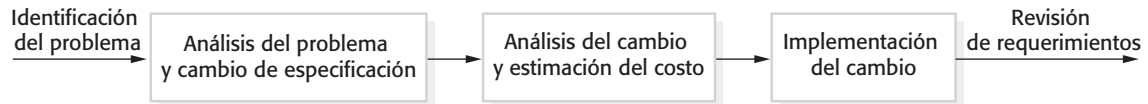
Una vez que se instala un sistema, y se utiliza con regularidad, surgirán inevitablemente nuevos requerimientos. Es difícil que los usuarios y clientes del sistema anticipen qué efectos tendrá el nuevo sistema sobre sus procesos de negocios y la forma en que se hace el trabajo. Una vez que los usuarios finales experimentan el sistema, descubrirán nuevas necesidades y prioridades. Existen muchas razones por las que es inevitable el cambio:

1. Los ambientes empresarial y técnico del sistema siempre cambian después de la instalación. Puede introducirse nuevo hardware, y quizá sea necesario poner en interfaz el sistema con otros sistemas, cambiar las prioridades de la empresa (con los consecuentes cambios en el sistema de apoyo requerido) e introducir nuevas leyes y regulaciones que el sistema deba cumplir cabalmente.
2. Los individuos que pagan por un sistema y los usuarios de dicho sistema, por lo general, no son los mismos. Los clientes del sistema imponen requerimientos debido a restricciones organizativas y presupuestales. Esto podría estar en conflicto con los requerimientos del usuario final y, después de la entrega, probablemente deban agregarse nuevas características para apoyar al usuario, si el sistema debe cubrir sus metas.
3. Los sistemas grandes tienen regularmente una comunidad de usuarios diversa, en la cual muchos individuos tienen diferentes requerimientos y prioridades que quizás estén en conflicto o sean contradictorios. Los requerimientos finales del sistema inevitablemente tienen un compromiso entre sí y, con la experiencia, a menudo se descubre que el equilibrio de apoyo brindado a diferentes usuarios tiene que cambiar.

La administración de requerimientos es el proceso de comprender y controlar los cambios en los requerimientos del sistema. Es necesario seguir la pista de requerimientos individuales y mantener los vínculos entre los requerimientos dependientes, de manera que pueda valorarse el efecto del cambio en los requerimientos. También es preciso establecer un proceso formal para hacer cambios a las propuestas y vincular éstos con los requerimientos del sistema. El proceso formal de la administración de requerimientos debe comenzar tan pronto como esté disponible un borrador del documento de requerimientos. Sin embargo, hay que empezar a planear cómo administrar el cambio en los requerimientos durante el proceso de adquisición de los mismos.

#### 4.7.1 Planeación de la administración de requerimientos

La planeación es una primera etapa esencial en el proceso de administración de requerimientos. Esta etapa establece el nivel de detalle que se requiere en la administración de requerimientos. Durante la etapa de administración de requerimientos, usted tiene que decidir sobre:



**Figura 4.18**  
Administración  
del cambio de  
requerimientos

1. *Identificación de requerimientos* Cada requerimiento debe identificarse de manera exclusiva, de forma que pueda tener referencia cruzada con otros requerimientos y usarse en las evaluaciones de seguimiento.
2. *Un proceso de administración del cambio* Éste es el conjunto de actividades que valoran el efecto y costo de los cambios. En la siguiente sección se estudia con más detalle este proceso.
3. *Políticas de seguimiento* Dichas políticas definen las relaciones entre cada requerimiento, así como entre los requerimientos y el diseño del sistema que debe registrarse. La política de seguimiento también tiene que definir cómo mantener dichos registros.
4. *Herramientas de apoyo* La administración de requerimientos incluye el procesamiento de grandes cantidades de información acerca de los requerimientos. Las herramientas disponibles varían desde sistemas especializados de administración de requerimientos, hasta hojas de cálculo y sistemas de bases de datos simples.

La administración de requerimientos necesita apoyo automatizado y herramientas de software, para lo cual deben seleccionarse durante la fase de planeación. Se necesitan herramientas de apoyo para:

1. *Almacenamiento de requerimientos* Los requerimientos tienen que mantenerse en un almacén de datos administrado y seguro, que sea accesible para todos quienes intervienen en el proceso de ingeniería de requerimientos.
2. *Administración del cambio* El proceso de administración del cambio (figura 4.18) se simplifica si está disponible la herramienta de apoyo activa.
3. *Administración del seguimiento* Como se estudió anteriormente, la herramienta de apoyo para el seguimiento permite la identificación de requerimientos relacionados. Algunas herramientas que están disponibles usan técnicas de procesamiento en lenguaje natural, para ayudar a descubrir posibles relaciones entre los requerimientos.

Para sistemas pequeños, quizá no sea necesario usar herramientas especializadas de administración de requerimientos. El proceso de administración de requerimientos puede apoyarse con el uso de funciones disponibles en procesadores de texto, hojas de cálculo y bases de datos de PC. Sin embargo, para sistemas más grandes se requieren herramientas de apoyo más especializadas. En las páginas Web del libro se incluyen vínculos a información acerca de herramientas de administración de requerimientos.

#### 4.7.2 Administración del cambio en los requerimientos

La administración del cambio en los requerimientos (figura 4.18) debe aplicarse a todos los cambios propuestos a los requerimientos de un sistema, después de aprobarse el documento de requerimientos. La administración del cambio es esencial porque es necesario determinar si los beneficios de implementar nuevos requerimientos están justificados por



### Seguimiento de requerimientos

Es necesario seguir la huella de las relaciones entre requerimientos, sus fuentes y el diseño del sistema, de modo que usted pueda analizar las razones para los cambios propuestos, así como el efecto que dichos cambios tengan probablemente sobre otras partes del sistema. Es necesario poder seguir la pista de cómo un cambio se propaga hacia el sistema. ¿Por qué?

<http://www.SoftwareEngineering-9.com/Web/Requirements/ReqTraceability.html>

los costos de la implementación. La ventaja de usar un proceso formal para la administración del cambio es que todas las propuestas de cambio se tratan de manera consistente y los cambios al documento de requerimientos se realizan en una forma controlada.

Existen tres etapas principales de un proceso de administración del cambio:

1. *Análisis del problema y especificación del cambio* El proceso comienza con la identificación de un problema en los requerimientos o, en ocasiones, con una propuesta de cambio específica. Durante esta etapa, el problema o la propuesta de cambio se analizan para comprobar que es válida. Este análisis retroalimenta al solicitante del cambio, quien responderá con una propuesta de cambio de requerimientos más específica, o decidirá retirar la petición.
2. *Análisis del cambio y estimación del costo* El efecto del cambio propuesto se valora usando información de seguimiento y conocimiento general de los requerimientos del sistema. El costo por realizar el cambio se estima en términos de modificaciones al documento de requerimientos y, si es adecuado, al diseño y la implementación del sistema. Una vez completado este análisis, se toma una decisión acerca de si se procede o no con el cambio de requerimientos.
3. *Implementación del cambio* Se modifican el documento de requerimientos y, donde sea necesario, el diseño y la implementación del sistema. Hay que organizar el documento de requerimientos de forma que sea posible realizar cambios sin reescritura o reorganización extensos. Conforme a los programas, la variabilidad en los documentos se logra al minimizar las referencias externas y al hacer las secciones del documento tan modulares como sea posible. De esta manera, secciones individuales pueden modificarse y sustituirse sin afectar otras partes del documento.

Si un nuevo requerimiento tiene que implementarse urgentemente, siempre existe la tentación para cambiar el sistema y luego modificar de manera retrospectiva el documento de requerimientos. Hay que tratar de evitar esto, pues casi siempre conducirá a que la especificación de requerimientos y la implementación del sistema se salgan de ritmo. Una vez realizados los cambios al sistema, es fácil olvidar la inclusión de dichos cambios en el documento de requerimientos, o bien, agregar información al documento de requerimientos que sea inconsistente con la implementación.

Los procesos de desarrollo ágil, como la programación extrema, se diseñaron para enfrentar los requerimientos que cambian durante el proceso de desarrollo. En dichos procesos, cuando un usuario propone un cambio de requerimientos, éste no pasa por un proceso de administración del cambio formal. En vez de ello, el usuario tiene que priorizar dicho cambio y, si es de alta prioridad, decidir qué características del sistema planeadas para la siguiente iteración pueden eliminarse.

## PUNTOS CLAVE

- Los requerimientos para un sistema de software establecen lo que debe hacer el sistema y definen las restricciones sobre su operación e implementación.
- Los requerimientos funcionales son enunciados de los servicios que debe proporcionar el sistema, o descripciones de cómo deben realizarse algunos cálculos.
- Los requerimientos no funcionales restringen con frecuencia el sistema que se va a desarrollar y el proceso de desarrollo a usar. Éstos pueden ser requerimientos del producto, requerimientos organizacionales o requerimientos externos. A menudo se relacionan con las propiedades emergentes del sistema y, por lo tanto, se aplican al sistema en su conjunto.
- El documento de requerimientos de software es un enunciado acordado sobre los requerimientos del sistema. Debe organizarse de forma que puedan usarlo tanto los clientes del sistema como los desarrolladores del software.
- El proceso de ingeniería de requerimientos incluye un estudio de factibilidad, adquisición y análisis de requerimientos, especificación de requerimientos, validación de requerimientos y administración de requerimientos.
- La adquisición y el análisis de requerimientos es un proceso iterativo que se representa como una espiral de actividades: descubrimiento de requerimientos, clasificación y organización de requerimientos, negociación de requerimientos y documentación de requerimientos.
- La validación de requerimientos es el proceso de comprobar la validez, la consistencia, la totalidad, el realismo y la verificabilidad de los requerimientos.
- Los cambios empresariales, organizacionales y técnicos conducen inevitablemente a cambios en los requerimientos para un sistema de software. La administración de requerimientos es el proceso de gestionar y controlar dichos cambios.

## LECTURAS SUGERIDAS

*Software Requirements, 2nd edition.* Este libro, diseñado para escritores y usuarios de requerimientos, analiza las buenas prácticas en la ingeniería de requerimientos. (K. M. Weigers, 2003, Microsoft Press.)

“Integrated requirements engineering: A tutorial”. Se trata de un ensayo de tutoría en el que se analizan las actividades de la ingeniería de requerimientos y cómo pueden adaptarse para ajustarse a las prácticas modernas de la ingeniería de software. (I. Sommerville, IEEE Software, 22(1), Jan–Feb 2005.) <http://dx.doi.org/10.1109/MS.2005.13>.

*Mastering the Requirements Process, 2nd edition.* Un libro bien escrito, fácil de leer, que se basa en un método particular (VOLERE), pero que también incluye múltiples buenos consejos generales acerca de la ingeniería de requerimientos. (S. Robertson y J. Robertson, 2006, Addison-Wesley.)

“Research Directions in Requirements Engineering”. Un buen estudio de la investigación en ingeniería de requerimientos que destaca los futuros retos en la investigación en el área, con la finalidad de enfrentar conflictos como la escala y la agilidad. (B. H. C. Cheng y J. M. Atlee, Proc. Conf on Future of Software Engineering, IEEE Computer Society, 2007.) <http://dx.doi.org/10.1109/FOSE.2007.17>.

## EJERCICIOS

- 4.1. Identifique y describa brevemente cuatro tipos de requerimientos que puedan definirse para un sistema basado en computadora.
- 4.2. Descubra las ambigüedades u omisiones en el siguiente enunciado de requerimientos de un sistema de emisión de boletos:

Un sistema automatizado de emisión de boletos vende boletos de ferrocarril. Los usuarios seleccionan su destino e ingresan un número de tarjeta de crédito y uno de identificación personal. El boleto de ferrocarril se emite y se carga en su cuenta de tarjeta de crédito. Cuando el usuario oprime el botón *start*, se activa una pantalla de menú con los posibles destinos, junto con un mensaje que pide al usuario seleccionar un destino. Una vez seleccionado el destino, se solicita a los usuarios ingresar su tarjeta de crédito. Se comprueba su validez y luego se pide al usuario ingresar un identificador personal. Cuando se valida la transacción crediticia, se emite el boleto.
- 4.3. Vuelva a escribir la descripción anterior usando el enfoque estructurado referido en este capítulo. Resuelva las ambigüedades identificadas de forma adecuada.
- 4.4. Escriba un conjunto de requerimientos no funcionales para el sistema de emisión de boletos, y establezca su fiabilidad y tiempo de respuesta esperados.
- 4.5. Con la técnica aquí sugerida, en que las descripciones en lenguaje natural se presentan en formato estándar, escriba requerimientos de usuario plausibles para las siguientes funciones:
  - Un sistema de bombeo de petróleo (gasolina) no asistido que incluya un lector de tarjeta de crédito. El cliente pasa la tarjeta en el lector, luego especifica la cantidad de combustible requerido. Se suministra el combustible y se deduce de la cuenta del cliente.
  - La función de dispensar efectivo en un cajero automático.
  - La función de revisión y corrección ortográfica en un procesador de textos.
- 4.6. Sugiera cómo un ingeniero responsable de redactar una especificación de requerimientos de sistema puede seguir la huella de las relaciones entre requerimientos funcionales y no funcionales.
- 4.7. Con su conocimiento de cómo se usa un cajero automático, desarrolle un conjunto de casos de uso que pudieran servir como base para comprender los requerimientos para el sistema de un cajero automático.
- 4.8. ¿Quién debería involucrarse en una revisión de requerimientos? Dibuje un modelo del proceso que muestre cómo podría organizarse una revisión de requerimientos.
- 4.9. Cuando tienen que hacerse cambios de emergencia a los sistemas, es posible que deba modificarse el software del sistema antes de aprobar los cambios a los requerimientos. Sugiera un modelo de un proceso para realizar dichas modificaciones, que garantice que el documento de requerimientos y la implementación del sistema no serán inconsistentes.
- 4.10. Usted acepta un empleo con un usuario de software, quien contrató a su empleador anterior con la finalidad de desarrollar un sistema para ellos. Usted descubre que la interpretación de los requerimientos de su compañía es diferente de la interpretación tomada por su antiguo empleador. Discuta qué haría en tal situación. Usted sabe que los costos para su actual empleador aumentarán si no se resuelven las ambigüedades. Sin embargo, también tiene una responsabilidad de confidencialidad con su empleador anterior.

## REFERENCIAS

- Beck, K. (1999). "Embracing Change with Extreme Programming". *IEEE Computer*, **32** (10), 70–8.
- Crabtree, A. (2003). *Designing Collaborative Systems: A Practical Guide to Ethnography*. London: Springer-Verlag.
- Davis, A. M. (1993). *Software Requirements: Objects, Functions and States*. Englewood Cliffs, NJ: Prentice Hall.
- IEEE. (1998). "IEEE Recommended Practice for Software Requirements Specifications". En *IEEE Software Engineering Standards Collection*. Los Alamitos, Ca.: IEEE Computer Society Press.
- Jacobson, I., Christerson, M., Jonsson, P. y Overgaard, G. (1993). *Object-Oriented Software Engineering*. Wokingham: Addison-Wesley.
- Kotonya, G. y Sommerville, I. (1998). *Requirements Engineering: Processes and Techniques*. Chichester, UK: John Wiley and Sons.
- Larman, C. (2002). *Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and the Unified Process*. Englewood Cliff, NJ: Prentice Hall.
- Martin, D., Rodden, T., Rouncefield, M., Sommerville, I. y Viller, S. (2001). "Finding Patterns in the Fieldwork". *Proc. ECSCW'01*. Bonn: Kluwer. 39–58.
- Martin, D., Rouncefield, M. y Sommerville, I. (2002). "Applying patterns of interaction to work (re) design: E-government and planning". *Proc. ACM CHI'2002*, ACM Press. 235–42.
- Martin, D. y Sommerville, I. (2004). "Patterns of interaction: Linking ethnomethodology and design". *ACM Trans. on Computer-Human Interaction*, **11** (1), 59–89.
- Robertson, S. y Robertson, J. (1999). *Mastering the Requirements Process*. Harlow, UK: Addison-Wesley.
- Sommerville, I., Rodden, T., Sawyer, P., Bentley, R. y Twidale, M. (1993). "Integrating ethnography into the requirements engineering process". *Proc. RE'93*, San Diego CA.: IEEE Computer Society Press. 165–73.
- Stevens, P. y Pooley, R. (2006). *Using UML: Software Engineering with Objects and Components*, 2nd ed. Harlow, UK: Addison Wesley.
- Suchman, L. (1987). *Plans and Situated Actions*. Cambridge: Cambridge University Press.
- Viller, S. y Sommerville, I. (1999). "Coherence: An Approach to Representing Ethnographic Analyses in Systems Design". *Human-Computer Interaction*, **14** (1 & 2), 9–41.
- Viller, S. y Sommerville, I. (2000). "Ethnographically informed analysis for software engineers". *Int. J. of Human-Computer Studies*, **53** (1), 169–96.