

EL ROL DEL ANALISTA DE SISTEMAS

1

OBJETIVOS DE APRENDIZAJE

Una vez que haya dominado el material de este capítulo, podrá:

1. Recordar los tipos básicos de sistemas de cómputo con los que debe trabajar un analista de sistemas.
2. Entender la manera en que las nuevas tecnologías influyen en la dinámica de un sistema.
3. Reconocer los diversos roles de un analista de sistemas.
4. Conocer los pasos del SDLC y saber cómo aplicarlos a un sistema real.
5. Comprender la función de las herramientas CASE y cómo ayudan a un analista de sistemas.
6. Explorar otras metodologías como el diseño de sistemas orientados a objetos y la elaboración de prototipos.

Desde hace mucho tiempo, las organizaciones han reconocido la importancia de administrar recursos clave como la mano de obra y las materias primas. En la actualidad, la información se ha ganado el legítimo derecho de ser considerada como un recurso clave. Los encargados de la toma de decisiones por fin han comprendido que la información no es tan sólo un producto derivado de la conducción de los negocios, sino un impulsor de los mismos y que puede constituir un factor crucial en el éxito o fracaso de una empresa.

Para maximizar la utilidad de la información, una empresa debe administrarla de manera eficiente, como lo hace con los demás recursos. Los administradores deben comprender que los costos tienen una estrecha relación con la producción, distribución, seguridad, almacenamiento y recuperación de toda la información. A pesar de que la información está en todas partes, no es gratuita, y no se debe asumir que se podrá usar estratégicamente para aumentar la competitividad de una empresa.

La amplia disponibilidad de computadoras en red, junto con el acceso a Internet y la World Wide Web, han propiciado una explosión de la información en la sociedad en general y en los negocios en particular. La administración de la información generada por computadora difiere en aspectos importantes del manejo de los datos producidos por medios manuales. Por lo general hay una mayor cantidad de información de computadora por manejar. Los costos de organizarla y darle mantenimiento se pueden incrementar a niveles alarmantes, y con frecuencia los usuarios la consideran más precisa que la información obtenida por otros medios. En este capítulo se examinan los aspectos básicos de los diferentes tipos de sistemas de información, los diversos roles de los analistas de sistemas, las fases del ciclo de vida del desarrollo de sistemas (SDLC, *Systems Development Life Cycle*) y se presentan las herramientas de Ingeniería de Software Asistida por Computadora (CASE, *Computer-Aided Software Engineering*).

TIPOS DE SISTEMAS

Los sistemas de información se desarrollan con diversos propósitos, según las necesidades de la empresa. Los sistemas de procesamiento de transacciones (TPS, *Transaction Processing Systems*) funcionan al nivel operativo de una organización, los sistemas de automatización de la oficina (OAS, *Office Automation Systems*) y los sistemas de trabajo del conocimiento (KWS, *Knowledge Work Systems*) apoyan el trabajo al nivel del conocimiento. Los sistemas de información gerencial (MIS, *Management Information Systems*) y los sistemas de apoyo a la toma de decisiones (DSS, *Decision Support Systems*) se encuentran entre los sistemas de alto nivel. Los sistemas expertos aplican el conocimiento de los encargados de la toma de decisiones para solucionar problemas estructurados específicos. Los sistemas de apoyo a ejecutivos (ESS, *Executive Support Systems*) se encuentran en el nivel estratégico de la administración. Los sistemas de apoyo a la toma de decisiones en grupo (GDSS, *Group Decision Support Systems*) y los sistemas de trabajo corporativo apoyados por computadora (CSCWS, *Computer-Supported Collaborative Work Systems*), descritos de manera más general, auxilian la toma de decisiones semiestructuradas o no estructuradas a nivel de grupo.

En la figura 1.1 se muestra la diversidad de sistemas de información que podrían desarrollar los analistas. Observe que en la figura estos sistemas se representan de abajo hacia arriba, indicando que los TPS apoyan el nivel operativo, o más bajo, de la organización, mientras que los ESS, GDSS y CSCWS soportan el nivel estratégico, o más alto, apoyando la toma de decisiones semiestructuradas o las no estructuradas. En este libro se emplean de manera indistinta los términos *sistemas de información gerencial*, *sistemas de información* (IS, *Information Systems*), *sistemas de información computarizados* y *sistemas de información de negocios computarizados*, para denotar sistemas de información computarizados que apoyan el rango de actividades de negocios más amplio mediante la información que producen.

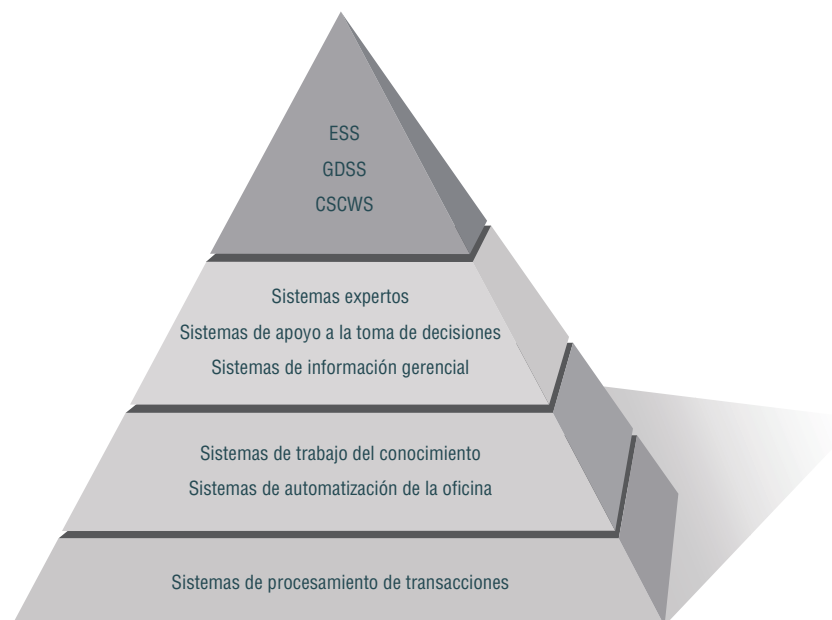
SISTEMAS DE PROCESAMIENTO DE TRANSACCIONES

Los sistemas de procesamiento de transacciones (TPS, *Transaction Processing Systems*) son sistemas de información computarizada creados para procesar grandes cantidades de datos relacionadas con transacciones rutinarias de negocios, como las nóminas y los inventarios. Un TPS elimina el fastidio que representa la realización de transacciones operativas necesarias y reduce el tiempo que una vez fue requerido para llevarlas a cabo de manera manual, aunque los usuarios aún tienen que capturar datos en los sistemas computarizados.

Los sistemas de procesamiento de transacciones expanden los límites de la organización dado que le permiten interactuar con entornos externos. Es importante para las operaciones cotidianas de un negocio, que estos sistemas funcionen sin ningún tipo de interrupción, puesto

FIGURA 1.1

Un analista de sistemas podría trabajar con alguno de estos sistemas o con todos ellos.



que los administradores recurren a los datos producidos por los TPS con el propósito de obtener información actualizada sobre el funcionamiento de sus empresas.

SISTEMAS DE AUTOMATIZACIÓN DE LA OFICINA Y SISTEMAS DE TRABAJO DEL CONOCIMIENTO

Existen dos clases de sistemas en el nivel del conocimiento de una organización. Los sistemas de automatización de la oficina (OAS, *Office Automation Systems*) apoyan a los trabajadores de datos, quienes por lo general no generan conocimientos nuevos, sino más bien analizan la información con el propósito de transformar los datos o manipularlos de alguna manera antes de compartirlos o, en su caso, distribuirlos formalmente con el resto de la organización y en ocasiones más allá de ésta. Entre los componentes más comunes de un OAS están el procesamiento de texto, las hojas de cálculo, la autoedición, la calendarización electrónica y las comunicaciones mediante correo de voz, correo electrónico y videoconferencia.

Los sistemas de trabajo del conocimiento (KWS, *Knowledge Work Systems*) sirven de apoyo a los trabajadores profesionales, como los científicos, ingenieros y médicos, en sus esfuerzos de creación de nuevo conocimiento y dan a éstos la posibilidad de compartirlo con sus organizaciones o con la sociedad.

SISTEMAS DE INFORMACIÓN GERENCIAL

Los sistemas de información gerencial (MIS, *Management Information Systems*) no reemplazan a los sistemas de procesamiento de transacciones, más bien, incluyen el procesamiento de transacciones. Los MIS son sistemas de información computarizados cuyo propósito es contribuir a la correcta interacción entre los usuarios y las computadoras. Debido a que requieren que los usuarios, el software (los programas de cómputo) y el hardware (las computadoras, impresoras, etc.), funcionen de manera coordinada, los sistemas de información gerencial dan apoyo a un espectro de tareas organizacionales mucho más amplio que los sistemas de procesamiento de transacciones, como el análisis y la toma de decisiones.

Para acceder a la información, los usuarios de un sistema de información gerencial comparten una base de datos común. Ésta almacena datos y modelos que ayudan al usuario a interpretar y aplicar los datos. Los sistemas de información gerencial producen información que se emplea en la toma de decisiones. Un sistema de información gerencial también puede contribuir a unificar algunas de las funciones de información computarizadas de una empresa, a pesar de que no existe como una estructura individual en ninguna parte de ésta.

SISTEMAS DE APOYO A LA TOMA DE DECISIONES

Los sistemas de apoyo a la toma de decisiones (DSS, *Decision Support Systems*) constituyen una clase de alto nivel de sistemas de información computarizada. Los DSS coinciden con los sistemas de información gerencial en que ambos dependen de una base de datos para abastecerse de datos. Sin embargo, difieren en que el DSS pone énfasis en el apoyo a la toma de decisiones en todas sus fases, aunque la decisión definitiva es responsabilidad exclusiva del encargado de tomarla. Los sistemas de apoyo a la toma de decisiones se ajustan más al gusto de la persona o grupo que los utiliza que a los sistemas de información gerencial tradicionales. En ocasiones se hace referencia a ellos como sistemas que se enfocan en la inteligencia de negocios.

SISTEMAS EXPERTOS E INTELIGENCIA ARTIFICIAL

La inteligencia artificial (AI, *Artificial Intelligence*) se puede considerar como el campo general para los sistemas expertos. La motivación principal de la AI ha sido desarrollar máquinas que tengan un comportamiento inteligente. Dos de las líneas de investigación de la AI son la comprensión del lenguaje natural y el análisis de la capacidad para razonar un problema hasta su conclusión lógica. Los sistemas expertos utilizan las técnicas de razonamiento de la AI para solucionar los problemas que les plantean los usuarios de negocios (y de otras áreas).

Los sistemas expertos conforman una clase muy especial de sistema de información que se ha puesto a disposición de usuarios de negocios gracias a la amplia disponibilidad de

hardware y software como computadoras personales (PCs) y generadores de sistemas expertos. Un sistema experto (también conocido como sistema basado en el conocimiento) captura y utiliza el conocimiento de un experto para solucionar un problema específico en una organización. Observe que a diferencia de un DSS, que cede al responsable la toma de la decisión definitiva, un sistema experto selecciona la mejor solución para un problema o una clase específica de problemas.

Los componentes básicos de un sistema experto son la base de conocimientos, un motor de inferencia que conecta al usuario con el sistema mediante el procesamiento de consultas realizadas con lenguajes como SQL (*Structured Query Language*, lenguaje de consultas estructurado) y la interfaz de usuario. Profesionales conocidos como ingenieros de conocimiento capturan la pericia de los expertos, construyen un sistema de cómputo que contiene este conocimiento experto y lo implementan. Es muy factible que la construcción e implementación de sistemas expertos se constituya en el trabajo futuro de muchos analistas de sistemas.

SISTEMAS DE APOYO A LA TOMA DE DECISIONES EN GRUPO Y SISTEMAS DE TRABAJO COLABORATIVO APOYADOS POR COMPUTADORA

Cuando los grupos requieren trabajar en conjunto para tomar decisiones semiestructuradas o no estructuradas, un sistema de apoyo a la toma de decisiones en grupo (GDSS, *Group Decision Support System*) podría ser la solución. Este tipo de sistemas, que se utilizan en salones especiales equipados con diversas configuraciones, faculta a los miembros del grupo a interactuar con apoyo electrónico —casi siempre software especializado— y la asistencia de un facilitador especial. Los sistemas de apoyo a la toma de decisiones en grupo tienen el propósito de unir a un grupo en la búsqueda de la solución a un problema con la ayuda de diversas herramientas como los sondeos, los cuestionarios, la lluvia de ideas y la creación de escenarios. El software GDSS puede diseñarse con el fin de minimizar las conductas negativas de grupo comunes, como la falta de participación originada por el miedo a las represalias si se expresa un punto de vista impopular o contrario, el control por parte de miembros elocuentes del grupo y la toma de decisiones conformista. En ocasiones se hace referencia a los GDSS con el término más general *sistemas de trabajo colaborativo apoyados por computadora* (CSCWS, *Computer-Supported Collaborative Work Systems*), que pueden contener el respaldo de un tipo de software denominado *groupware* para la colaboración en equipo a través de computadoras conectadas en red.

SISTEMAS DE APOYO A EJECUTIVOS

Cuando los ejecutivos recurren a la computadora, por lo general lo hacen en busca de métodos que los auxilien en la toma de decisiones de nivel estratégico. Los sistemas de apoyo a ejecutivos (ESS, *Executive Support Systems*) ayudan a estos últimos a organizar sus actividades relacionadas con el entorno externo mediante herramientas gráficas y de comunicaciones, que por lo general se encuentran en salas de juntas o en oficinas corporativas personales. A pesar de que los ESS dependen de la información producida por los TPS y los MIS, ayudan a los usuarios a resolver problemas de toma de decisiones no estructuradas, que no tienen una aplicación específica, mediante la creación de un entorno que contribuye a pensar en problemas estratégicos de una manera bien informada. Los ESS amplían y apoyan las capacidades de los ejecutivos al darles la posibilidad de comprender sus entornos.

INTEGRACIÓN DE LAS TECNOLOGÍAS DE SISTEMAS

Como se aprecia en la figura 1.2, a medida que se adopten y difundan las nuevas tecnologías, parte del trabajo de los analistas de sistemas se dedicará a la integración de los sistemas tradicionales con los nuevos. En esta sección se describen algunas de las nuevas tecnologías de información que los analistas de sistemas utilizarán para empresas que buscan integrar sus aplicaciones de comercio electrónico con sus negocios tradicionales, o bien, iniciar negocios electrónicos completamente nuevos.

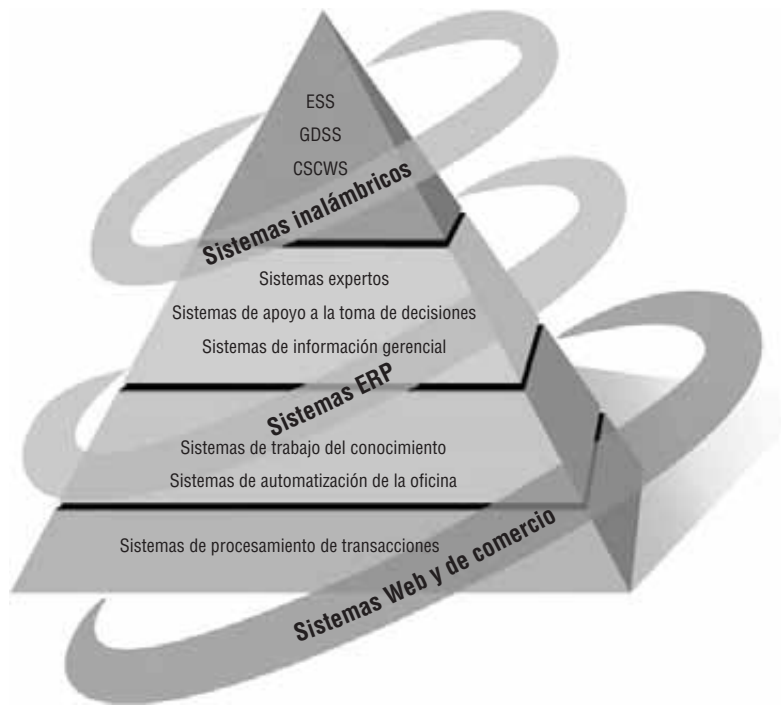


FIGURA 1.2

Los analistas de sistemas deben estar conscientes de que la integración de tecnologías abarca todos los tipos de sistemas.

APLICACIONES DE COMERCIO ELECTRÓNICO Y SISTEMAS WEB

Muchos de los sistemas que se describen en este libro pueden dotarse de una mayor funcionalidad si se migran a la World Wide Web o si desde su concepción se implementan como tecnologías basadas en la Web. En una encuesta reciente la mitad de todas las empresas pequeñas y medianas respondieron que Internet fue su estrategia preferida para buscar el crecimiento de sus negocios. Esta respuesta duplicó a la de aquellos que manifestaron su inclinación por realizar alianzas estratégicas como medio para crecer. Hay muchos beneficios derivados de la implementación de una aplicación en la Web:

1. Una creciente difusión de la disponibilidad de un servicio, producto, industria, persona o grupo.
2. La posibilidad de que los usuarios accedan las 24 horas.
3. La estandarización del diseño de la interfaz.
4. La creación de un sistema que se puede extender a nivel mundial y llegar a gente en lugares remotos sin preocuparse por la zona horaria en que se encuentren.

SISTEMAS DE PLANEACIÓN DE RECURSOS EMPRESARIALES

Muchas organizaciones consideran los beneficios potenciales que se derivan de la integración de los diversos sistemas de información que existen en los diferentes niveles administrativos, con funciones dispares. Esta integración es precisamente el propósito de los sistemas de planeación de recursos empresariales (ERP, *Enterprise Resource Planning*). El establecimiento de los sistemas ERP implica un enorme compromiso y cambio por parte de la organización. Es común que los analistas de sistemas desempeñen el papel de asesores en los proyectos de ERP que utilizan software patentado. Entre el software más conocido de ERP se encuentran SAP, PeopleSoft y paquetes de Oracle y J.D. Edwards. Algunos de estos paquetes están diseñados para migrar a las empresas a la Web. Por lo general, los analistas y algunos usuarios requieren capacitación, apoyo técnico y mantenimiento por parte del fabricante para diseñar, instalar, dar mantenimiento, actualizar y utilizar de manera apropiada un paquete de ERP en particular.

SISTEMAS PARA DISPOSITIVOS INALÁMBRICOS Y PORTÁTILES

Los analistas tienen la exigencia de diseñar una gran cantidad de nuevos sistemas y aplicaciones, muchos de ellos para dispositivos inalámbricos y computadoras portátiles como la

popular serie de computadoras Palm y otros asistentes personales digitales (PDAs, *Personal Digital Assistants*). Además, los analistas podrían llegar a diseñar redes de comunicaciones estándar o inalámbricas que integren voz, video y correo electrónico en intranets para una organización o extranets para la industria. El comercio electrónico inalámbrico se conoce como comercio móvil o m-commerce.

Las redes inalámbricas de área local (WLANs, *Wireless Local Area Networks*), las redes de fidelidad inalámbrica, conocidas como WI-FI, y las redes inalámbricas personales que agrupan a muchos tipos de dispositivos dentro del estándar conocido como Bluetooth, constituyen sistemas cuyo diseño podrían solicitarle a usted en su función de analista. (Para ahondar en las redes inalámbricas, véase el capítulo 17.)

En un contexto más avanzado, al analista podría solicitársele el diseño de agentes inteligentes, software que puede ayudar a los usuarios a ejecutar tareas mediante el aprendizaje de las preferencias del usuario a través del tiempo y, a continuación, realizando alguna acción sobre éstas. Por ejemplo, en la tecnología de recepción automática, un agente inteligente podría buscar temas de interés para el usuario en la Web, sin necesidad de que éste lo solicite, después de observar durante algún tiempo los patrones de comportamiento del usuario en relación con la información.

Un ejemplo de este tipo de software es el que desarrolla Microsoft con base en la estadística bayesiana (donde se utilizan estadísticas para inferir probabilidades) y la teoría de la toma de decisiones, en conjunto con el monitoreo del comportamiento de un usuario que maneja información entrante (como un mensaje de su casa, una llamada telefónica de un cliente, una llamada de celular o el análisis actualizado de su cartera de acciones). El resultado es software de manejo de notificaciones que da un valor monetario a cada pieza de información proveniente de diversas fuentes y también determina la mejor manera de desplegarla. Por ejemplo, con base en la teoría de la toma de decisiones, la probabilidad, la estadística y el propio comportamiento del usuario, a una llamada telefónica proveniente de la casa del usuario se le podría dar el valor de un peso y se desplegaría en la pantalla de la computadora, en tanto que a una llamada cuyo propósito es la venta de algún producto o servicio se le podría asignar el valor de 20 centavos (es decir, un valor inferior) y podría desplegarse como nota en un radiolocalizador.

SOFTWARE DE CÓDIGO ABIERTO

El software de código abierto es una alternativa al desarrollo de software tradicional cuyo código patentado se oculta a los usuarios. Representa un modelo de desarrollo y filosofía de distribución de software gratuito y publicación de su código fuente. Bajo este esquema, el código (las instrucciones para la computadora) se puede estudiar y compartir, y muchos usuarios y programadores tienen la posibilidad de modificarlo. Las convenciones que rigen a esta comunidad incluyen que todas las modificaciones que se hagan a un programa deben compartirse con todos aquellos que participan en el proyecto. Entre los ejemplos se encuentran el sistema operativo Linux y el software Apache empleado en servidores que alojan sitios Web.

Si el software es de distribución gratuita, ¿cómo ganan dinero las compañías? Para ello, tienen que proporcionar un servicio, personalizar programas para los usuarios y darles seguimiento con un soporte continuo. En un mundo de software de código abierto, el desarrollo de sistemas continuaría su evolución hacia una industria de servicios. Se apartaría del modelo de manufactura en el que los productos se licencian y empaquetan en cajas vistosas y se envían hasta nuestras puertas, al igual que cualquier otro producto manufacturado.

El desarrollo de código abierto es útil para los dispositivos portátiles y el equipo de comunicaciones. Su uso podría estimular el progreso en la creación de estándares para que los dispositivos se comunicaran con más facilidad. El uso generalizado del software de código abierto podría solucionar problemas que pudiera causar la escasez de programadores y algunos problemas complejos podrían resolverse mediante la colaboración de muchos especialistas.

LA NECESIDAD DEL ANÁLISIS Y DISEÑO DE SISTEMAS

El análisis y diseño de sistemas, tal como lo realizan los analistas de sistemas, tiene el propósito de analizar sistemáticamente la entrada o el flujo de datos, procesar o transformar da-



Y ésta es la razón por la cual necesitamos una computadora.

tos, el almacenamiento de datos y la salida de información en el contexto de una empresa en particular. Más aún, el análisis de sistemas se emplea para analizar, diseñar e implementar mejoras en el funcionamiento de las empresas, a través de sistemas de información computarizados.

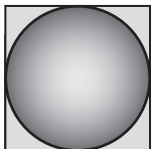
La instalación de un sistema sin una planeación adecuada conduce a una gran decepción y con frecuencia provoca que el sistema deje de utilizarse. El análisis y diseño de sistemas da forma al análisis y diseño de sistemas de información, un esfuerzo muy valioso que de otra manera podría haberse realizado de una manera fortuita. Se le puede considerar como una serie de procesos sistemáticamente emprendidos con el propósito de mejorar un negocio con ayuda de sistemas de información computarizados. Gran parte del análisis y diseño de sistemas implica trabajar con usuarios actuales y ocasionales de los sistemas de información.

Es importante que los usuarios intervengan de alguna manera durante el proyecto para completar con éxito los sistemas de información computarizados. Los analistas de sistemas, cuyos roles en la organización se describen a continuación, constituyen el otro componente esencial en el desarrollo de sistemas de información útiles.

ROLES DEL ANALISTA DE SISTEMAS

El analista de sistemas evalúa de manera sistemática el funcionamiento de un negocio mediante el examen de la entrada y el procesamiento de datos y su consiguiente producción de información, con el propósito de mejorar los procesos de una organización. Muchas mejoras incluyen un mayor apoyo a las funciones de negocios a través del uso de sistemas de información computarizados. Esta definición pone énfasis en un enfoque sistemático y metódico para analizar —y en consecuencia mejorar— lo que sucede en el contexto específico creado por un negocio.

Nuestra definición de analista de sistemas es amplia. El analista debe tener la capacidad de trabajar con todo tipo de gente y contar con suficiente experiencia en computadoras. El



CONTRATACIÓN SANA: SE SOLICITA AYUDA PARA COMERCIO ELECTRÓNICO

“Estarán felices de enterarse que logramos convencer a la administración de que debemos contratar un nuevo analista de sistemas que se especialice en el desarrollo de comercio electrónico”, comentó Al Falfa, analista de sistemas de la cadena internacional de tiendas Marathon Vitamin Shops. Actualmente se reúne con su numeroso equipo de analista de sistemas para determinar las cualidades con que debe contar el nuevo miembro de su equipo. Al continúa: “De hecho, mostraron tanto interés por la posibilidad de que nuestro equipo colabore en la estrategia de comercio electrónico de Marathon que me indicaron que comencemos de inmediato nuestra búsqueda por el especialista y no esperemos hasta el otoño”.

Ginger Rute, otra analista, muestra su aprobación: “Cuando la economía es saludable, la demanda de desarrolladores de sitios Web rebasa con mucho a la oferta. Debemos actuar con rapidez. Creo que el nuevo analista debe tener conocimientos en herramientas CASE, Visual Basic y JavaScript, por mencionar algunos”.

Al se sorprende al escuchar la larga lista de lenguajes que enumera Ginger y replica: “Tienes razón, ésa es una de nuestras opciones. Sin embargo, también me gustaría que el nuevo miembro tuviera algo de experiencia en negocios. La mayoría de los egresados de las escuelas tienen sólidos conocimientos de programación, pero también deberían saber sobre contabilidad, inventarios y distribución de bienes y servicios”.

La más nueva en el grupo de analistas de sistemas, Vita Minn, se incorpora al debate: “Una de las razones por las cuales me incliné a trabajar con todos ustedes fue porque considero que nos llevamos bastante bien unos con otros. Como tenía otras opciones, tuve cuidado de ver cómo era el ambiente aquí. Por lo que he visto, conformamos un grupo amistoso. Así que asegurémonos de contratar a alguien que cuente con una personalidad adecuada que se acople al equipo”.

Al está de acuerdo y continúa: “Vita tiene razón. El nuevo analista debe ser alguien que se comunique bien con nosotros, lo mismo que con

los clientes de negocios. Siempre nos estamos comunicando de alguna manera, ya sea mediante presentaciones formales, dibujando diagramas o entrevistando a los usuarios. Si entienden por qué se toman las decisiones, su trabajo también se facilitará. Asimismo, Marathon tiene interés en integrar el comercio electrónico en toda la empresa. Requerimos alguien que comprenda al menos la importancia estratégica de la Web. El diseño de páginas es sólo una pequeña parte de esto”.

Ginger interviene nuevamente con una buena dosis de sentido práctico: “Deja eso en manos de la administración. Sigo creyendo que la nueva persona debe ser un buen programador”. Luego reflexiona en voz alta: “¿Me pregunto qué tan importante será saber UML para el puesto?”

Después de escuchar con paciencia los argumentos de todos, uno de los analistas veteranos, Cal Siem, interviene, bromeando: “¡Mejor deberíamos ver si Superman está disponible!”

Mientras todos ríen, Al vislumbra la oportunidad de lograr el consenso y dice: “Hemos tenido la oportunidad de escuchar diferentes cualidades. Tomemos un momento y cada quien haga una lista de las cualidades que considere esenciales en la persona que se encargará del desarrollo del comercio electrónico. Las expondremos y continuaremos el debate hasta que definamos a la persona con suficiente detalle y podamos elaborar un perfil para que el departamento de recursos humanos le dé seguimiento”.

¿Qué cualidades debe buscar el equipo de analistas de sistemas al contratar al nuevo miembro del equipo de desarrollo de comercio electrónico? ¿Es más importante el conocimiento de lenguajes específicos o contar con habilidad para aprender con rapidez lenguajes y paquetes de software? ¿Qué tan importante es que la persona que se contrate tenga algunos conocimientos básicos de negocios? ¿Deberían todos los miembros del equipo contar con habilidades y conocimientos idénticos? ¿Qué rasgos personales y carácter debe tener un analista de sistemas que trabaje en el desarrollo de comercio electrónico?

analista desempeña diversos roles, en ocasiones varios de ellos al mismo tiempo. Los tres roles principales del analista de sistemas son el de consultor, experto en soporte técnico y agente de cambio.

EL ROL DE CONSULTOR DEL ANALISTA DE SISTEMAS

Con frecuencia, el analista de sistemas desempeña el rol de consultor para un negocio y, por tanto, podría ser contratado de manera específica para enfrentar los problemas de sistemas de información de una empresa. Esta contratación se puede traducir en una ventaja porque los consultores externos tienen una perspectiva fresca de la cual carecen los demás miembros de una organización. También se puede traducir en una desventaja porque alguien externo nunca conocerá la verdadera cultura organizacional. En su función de consultor externo, usted dependerá en gran medida de los métodos sistemáticos que se explican en este libro para analizar y diseñar sistemas de información apropiados para una empresa en particular. Además, tendrá que apoyarse en los usuarios de los sistemas de información para entender la cultura organizacional desde la perspectiva que tienen ellos.

EL ROL DE EXPERTO EN SOPORTE TÉCNICO DEL ANALISTA DE SISTEMAS

Otro rol que tendrá que desempeñar es el de experto en soporte técnico dentro de la empresa en la cual labora de manera regular. En este rol el analista recurre a su experiencia

profesional con el hardware y software de cómputo y al uso que se le da en el negocio. Con frecuencia, este trabajo no implica un proyecto completo de sistemas, sino más bien la realización de pequeñas modificaciones o la toma de decisiones que se circunscriben a un solo departamento.

Como experto de soporte técnico, usted no está a cargo del proyecto; tan sólo actúa como recurso para aquellos que sí lo están. Si usted es un analista de sistemas contratado por una empresa de manufactura o servicios, gran parte de sus actividades podrían ajustarse a este rol.

EL ROL DE AGENTE DE CAMBIO DEL ANALISTA DE SISTEMAS

El rol más completo y de mayor responsabilidad que asume el analista de sistemas es el de agente de cambio, ya sea interno o externo para la empresa. Como analista, usted es un agente de cambio si desempeña cualquiera de las actividades relacionadas con el ciclo de vida del desarrollo de sistemas (que se explicará en la siguiente sección) y está presente en la empresa durante un largo periodo (de dos semanas a más de un año). Un agente de cambio se puede definir como alguien que sirve de catalizador para el cambio, desarrolla un plan para el cambio y coopera con los demás para facilitar el cambio.

Su presencia en el negocio inicia el cambio. Como analista de datos, usted debe estar consciente de este hecho y utilizarlo como punto de partida para su análisis. De ahí que tenga que interactuar con los usuarios y la administración (si no son uno solo y el mismo) desde el principio de su proyecto. Sin su colaboración usted no podría entender lo que ocurre en una organización y el cambio real nunca se daría.

Si el cambio (es decir, las mejoras al negocio que se pueden concretar mediante los sistemas de información) parece factible después de efectuar el análisis, el siguiente paso es desarrollar un plan para el cambio de manera conjunta con quienes tienen la facultad de autorizarlo. Una vez que se haya alcanzado el consenso acerca de los cambios por realizar, usted tendrá que interactuar constantemente con quienes vayan a cambiar.

En su calidad de analista de sistemas desempeñando la función de agente de cambio, debe promover un cambio que involucre el uso de los sistemas de información. También es parte de su tarea enseñar a los usuarios el proceso del cambio, ya que las modificaciones a un sistema de información no sólo afectan a éste sino que provocan cambios en el resto de la organización.

CUALIDADES DEL ANALISTA DE SISTEMAS

De las descripciones anteriores sobre los roles que desempeña el analista de sistemas, se deduce fácilmente que el analista exitoso debe contar con una amplia gama de cualidades. Hay una gran diversidad de personas trabajando como analistas de sistemas, por lo que cualquier descripción que intente ser general está destinada a quedarse corta en algún sentido. No obstante, la mayoría de los analistas de sistemas tienen algunas cualidades comunes.

En primer lugar, el analista es un solucionador de problemas. Es una persona que aborda como un reto el análisis de problemas y que disfruta al diseñar soluciones factibles. Cuando es necesario, el analista debe contar con la capacidad de afrontar sistemáticamente cualquier situación mediante la correcta aplicación de herramientas, técnicas y su experiencia. El analista también debe ser un comunicador con capacidad para relacionarse con los demás durante extensos periodos. Necesita suficiente experiencia en computación para programar, entender las capacidades de las computadoras, recabar los requisitos de información de los usuarios y comunicarlos a los programadores. Asimismo, debe tener una ética personal y profesional firme que le ayude a moldear las relaciones con sus clientes.

El analista de sistemas debe ser una persona autodisciplinada y automotivada, con la capacidad de administrar y coordinar los innumerables recursos de un proyecto, incluyendo a otras personas. La profesión de analista de sistemas es muy exigente; pero es una profesión en constante evolución que siempre trae nuevos retos.

EL CICLO DE VIDA DEL DESARROLLO DE SISTEMAS

A lo largo de este capítulo, nos hemos referido al enfoque sistemático que el analista toma en relación con el análisis y diseño de sistemas de información. Gran parte de este enfoque se incluye en el ciclo de vida del desarrollo de sistemas (SDLC, *Systems Development Life Cycle*). El SDLC es un enfoque por fases para el análisis y el diseño cuya premisa principal consiste en que los sistemas se desarrollan mejor utilizando un ciclo específico de actividades del analista y el usuario.

Los analistas no se ponen de acuerdo en la cantidad de fases que incluye el ciclo de vida del desarrollo de sistemas, pero en general alaban su enfoque organizado. Aquí hemos dividido el ciclo en siete fases, como se aprecia en la figura 1.3. A pesar de que cada fase se explica por separado, nunca se realiza como un paso aislado. Más bien, es posible que varias actividades ocurran de manera simultánea, y algunas de ellas podrían repetirse. Es más práctico considerar que el SDLC se realiza por fases (con actividades en pleno apogeo que se traslapan con otras hasta terminarse por completo) y no en pasos aislados.

IDENTIFICACIÓN DE PROBLEMAS, OPORTUNIDADES Y OBJETIVOS

En esta primera fase del ciclo de vida del desarrollo de sistemas, el analista se ocupa de identificar problemas, oportunidades y objetivos. Esta etapa es crítica para el éxito del resto del proyecto, pues a nadie le agrada desperdiciar tiempo trabajando en un problema que no era el que se debía resolver.

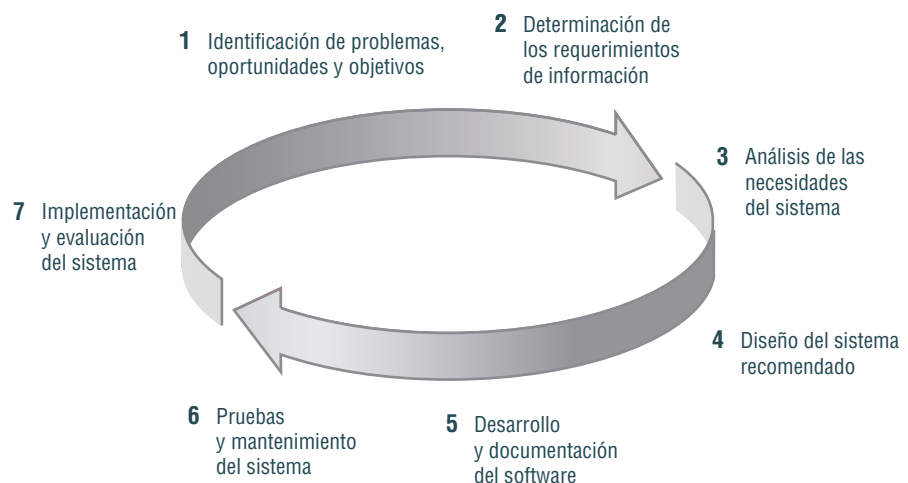
La primera fase requiere que el analista observe objetivamente lo que sucede en un negocio. A continuación, en conjunto con otros miembros de la organización, el analista determina con precisión cuáles son los problemas. Con frecuencia los problemas son detectados por alguien más, y ésta es la razón de la llamada inicial al analista. Las oportunidades son situaciones que el analista considera susceptibles de mejorar utilizando sistemas de información computarizados. El aprovechamiento de las oportunidades podría permitir a la empresa obtener una ventaja competitiva o establecer un estándar para la industria.

La identificación de objetivos también es una parte importante de la primera fase. En primer lugar, el analista debe averiguar lo que la empresa trata de conseguir. A continuación, podrá determinar si algunas funciones de las aplicaciones de los sistemas de información pueden contribuir a que el negocio alcance sus objetivos aplicándolas a problemas u oportunidades específicos.

Los usuarios, los analistas y los administradores de sistemas que coordinan el proyecto son los involucrados en la primera fase. Las actividades de esta fase consisten en entrevistar a los encargados de coordinar a los usuarios, sintetizar el conocimiento obtenido, estimar el alcance del proyecto y documentar los resultados. El resultado de esta fase es un informe de viabilidad que incluye una definición del problema y un resumen de los objetivos. A continuación, la administración debe decidir si se sigue adelante con el proyecto propuesto. Si el

FIGURA 1.3

Las siete fases del ciclo de vida del desarrollo de sistemas.



grupo de usuarios no cuenta con fondos suficientes, si desea atacar problemas distintos, o si la solución a estos problemas no amerita un sistema de cómputo, se podría sugerir una solución diferente y el proyecto de sistemas se cancelaría.

DETERMINACIÓN DE LOS REQUERIMIENTOS DE INFORMACIÓN

La siguiente fase que enfrenta el analista es la determinación de los requerimientos de información de los usuarios. Entre las herramientas que se utilizan para determinar los requerimientos de información de un negocio se encuentran métodos interactivos como las entrevistas, los muestreos, la investigación de datos impresos y la aplicación de cuestionarios; métodos que no interfieren con el usuario como la observación del comportamiento de los encargados de tomar las decisiones y sus entornos de oficina, al igual que métodos de amplio alcance como la elaboración de prototipos.

El desarrollo rápido de aplicaciones (RAD, *Rapid Application Development*) es un enfoque orientado a objetos para el desarrollo de sistemas que incluye un método de desarrollo (que abarca la generación de requerimientos de información) y herramientas de software. En este libro se aborda en el capítulo 6, en conjunto con la elaboración de prototipos, porque su enfoque filosófico es similar, aunque su método para crear un diseño con rapidez y obtener una pronta retroalimentación por parte de los usuarios es un poco diferente. (En el capítulo 18 se abunda en los enfoques orientados a objetos.)

En la fase de determinación de los requerimientos de información del SDLC, el analista se esfuerza por comprender la información que necesitan los usuarios para llevar a cabo sus actividades. Como puede ver, varios de los métodos para determinar los requerimientos de información implican interactuar directamente con los usuarios. Esta fase es útil para que el analista confirme la idea que tiene de la organización y sus objetivos. En ocasiones sólo realizan las dos primeras fases del ciclo de vida del desarrollo de sistemas. Esta clase de estudio podría tener un propósito distinto y por lo general la lleva a la práctica un especialista conocido como analista de información (IA, *Information Analyst*).

Los implicados en esta fase son el analista y los usuarios, por lo general trabajadores y gerentes del área de operaciones. El analista de sistemas necesita conocer los detalles de las funciones del sistema actual: el quién (la gente involucrada), el qué (la actividad del negocio), el dónde (el entorno donde se desarrollan las actividades), el cuándo (el momento oportuno) y el cómo (la manera en que se realizan los procedimientos actuales) del negocio que se estudia. A continuación el analista debe preguntar la razón por la cual se utiliza el sistema actual. Podría haber buenas razones para realizar los negocios con los métodos actuales, y es importante tomarlas en cuenta al diseñar un nuevo sistema.

Sin embargo, si la razón de ser de las operaciones actuales es que “siempre se han hecho de esta manera”, quizá será necesario que el analista mejore los procedimientos. La reingeniería de procesos de negocios podría ser útil para conceptualizar el negocio de una manera creativa. Al término de esta fase, el analista debe conocer el funcionamiento del negocio y poseer información muy completa acerca de la gente, los objetivos, los datos y los procedimientos implicados.

ANÁLISIS DE LAS NECESIDADES DEL SISTEMA

La siguiente fase que debe enfrentar el analista tiene que ver con el análisis de las necesidades del sistema. De nueva cuenta, herramientas y técnicas especiales auxilian al analista en la determinación de los requerimientos. Una de estas herramientas es el uso de diagramas de flujo de datos para graficar las entradas, los procesos y las salidas de las funciones del negocio en una forma gráfica estructurada. A partir de los diagramas de flujo de datos se desarrolla un diccionario de datos que enlista todos los datos utilizados en el sistema, así como sus respectivas especificaciones.

Durante esta fase el analista de sistemas analiza también las decisiones estructuradas que se hayan tomado. Las decisiones estructuradas son aquellas en las cuales se pueden determinar las condiciones, las alternativas de condición, las acciones y las reglas de acción. Existen tres métodos principales para el análisis de decisiones estructuradas: español estructurado, tablas y árboles de decisión.

En este punto del ciclo de vida del desarrollo de sistemas, el analista prepara una propuesta de sistemas que sintetiza sus hallazgos, proporciona un análisis de costo/beneficio de las alternativas y ofrece, en su caso, recomendaciones sobre lo que se debe hacer. Si la administración de la empresa considera factible alguna de las recomendaciones, el analista sigue adelante. Cada problema de sistemas es único, y nunca existe sólo una solución correcta. La manera de formular una recomendación o solución depende de las cualidades y la preparación profesional de cada analista.

DISEÑO DEL SISTEMA RECOMENDADO

En la fase de diseño del ciclo de vida del desarrollo de sistemas, el analista utiliza la información recopilada en las primeras fases para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos que aseguran que los datos que ingresen al sistema de información sean correctos. Además, el analista facilita la entrada eficiente de datos al sistema de información mediante técnicas adecuadas de diseño de formularios y pantallas.

La concepción de la interfaz de usuario forma parte del diseño lógico del sistema de información. La interfaz conecta al usuario con el sistema y por tanto es sumamente importante. Entre los ejemplos de interfaces de usuario se encuentran el teclado (para teclear preguntas y respuestas), los menús en pantalla (para obtener los comandos de usuario) y diversas interfaces gráficas de usuario (GUIs, *Graphical User Interfaces*) que se manejan a través de un ratón o una pantalla sensible al tacto.

La fase de diseño también incluye el diseño de archivos o bases de datos que almacenarán gran parte de los datos indispensables para los encargados de tomar las decisiones en la organización. Una base de datos bien organizada es el cimiento de cualquier sistema de información. En esta fase el analista también interactúa con los usuarios para diseñar la salida (en pantalla o impresa) que satisfaga las necesidades de información de estos últimos.

Finalmente, el analista debe diseñar controles y procedimientos de respaldo que protejan al sistema y a los datos, y producir paquetes de especificaciones de programa para los programadores. Cada paquete debe contener esquemas para la entrada y la salida, especificaciones de archivos y detalles del procesamiento; también podría incluir árboles o tablas de decisión, diagramas de flujo de datos, un diagrama de flujo de sistema, y los nombres y funciones de cualquier rutina de código previamente escrita.

DESARROLLO Y DOCUMENTACIÓN DEL SOFTWARE

En la quinta fase del ciclo de vida del desarrollo de sistemas, el analista trabaja de manera conjunta con los programadores para desarrollar cualquier software original necesario. Entre las técnicas estructuradas para diseñar y documentar software se encuentran los diagramas de estructura, los diagramas de Nassi-Shneiderman y el pseudocódigo. El analista se vale de una o más de estas herramientas para comunicar al programador lo que se requiere programar.

Durante esta fase el analista también trabaja con los usuarios para desarrollar documentación efectiva para el software, como manuales de procedimientos, ayuda en línea y sitios Web que incluyan respuestas a preguntas frecuentes (FAQ, *Frequently Asked Questions*) en archivos "Léame" que se integrarán en el nuevo software. La documentación indica a los usuarios cómo utilizar el software y lo que deben hacer en caso de que surjan problemas derivados de este uso.

Los programadores desempeñan un rol clave en esta fase porque diseñan, codifican y eliminan errores sintácticos de los programas de cómputo. Si el programa se ejecutará en un entorno de *mainframe*, se debe crear un lenguaje de control de trabajos (JCL, *Job Control Language*). Para garantizar la calidad, un programador podría efectuar un repaso estructurado del diseño o del código con el propósito de explicar las partes complejas del programa a otro equipo de programadores.

PRUEBA Y MANTENIMIENTO DEL SISTEMA

Antes de poner el sistema en funcionamiento es necesario probarlo. Es mucho menos costoso encontrar los problemas antes que el sistema se entregue a los usuarios. Una parte de las pruebas las realizan los programadores solos, y otra la llevan a cabo de manera conjunta con los analistas de sistemas. Primero se realiza una serie de pruebas con datos de muestra para determinar con precisión cuáles son los problemas y posteriormente se realiza otra con datos reales del sistema actual.

El mantenimiento del sistema de información y su documentación empiezan en esta fase y se llevan a cabo de manera rutinaria durante toda su vida útil. Gran parte del trabajo habitual del programador consiste en el mantenimiento, y las empresas invierten enormes sumas de dinero en esta actividad. Parte del mantenimiento, como las actualizaciones de programas, se pueden realizar de manera automática a través de un sitio Web. Muchos de los procedimientos sistemáticos que el analista emplea durante el ciclo de vida del desarrollo de sistemas pueden contribuir a garantizar que el mantenimiento se mantendrá al mínimo.

IMPLEMENTACIÓN Y EVALUACIÓN DEL SISTEMA

Ésta es la última fase del desarrollo de sistemas, y aquí el analista participa en la implementación del sistema de información. En esta fase se capacita a los usuarios en el manejo del sistema. Parte de la capacitación la imparten los fabricantes, pero la supervisión de ésta es responsabilidad del analista de sistemas. Además, el analista tiene que planear una conversión gradual del sistema anterior al actual. Este proceso incluye la conversión de archivos de formatos anteriores a los nuevos, o la construcción de una base de datos, la instalación de equipo y la puesta en producción del nuevo sistema.

Se menciona la evaluación como la fase final del ciclo de vida del desarrollo de sistemas principalmente en aras del debate. En realidad, la evaluación se lleva a cabo durante cada una de las fases. Un criterio clave que se debe cumplir es si los usuarios a quienes va dirigido el sistema lo están utilizando realmente.

Debe hacerse hincapié en que, con frecuencia, el trabajo de sistemas es cíclico. Cuando un analista termina una fase del desarrollo de sistemas y pasa a la siguiente, el surgimiento de un problema podría obligar al analista a regresar a la fase previa y modificar el trabajo realizado.

IMPACTO DEL MANTENIMIENTO

Después de instalar un sistema, se le debe dar mantenimiento, es decir, los programas de cómputo tienen que ser modificados y actualizados cuando lo requieran. En la figura 1.4 se ilustra el tiempo promedio que se invierte en darle mantenimiento a un MIS típico. Según estimaciones, los departamentos invierten en mantenimiento de 48 a 60 por ciento del tiempo total del desarrollo de sistemas. Queda muy poco tiempo para el desarrollo de nue-

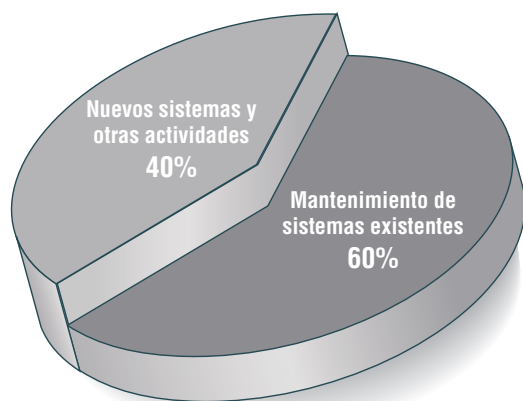
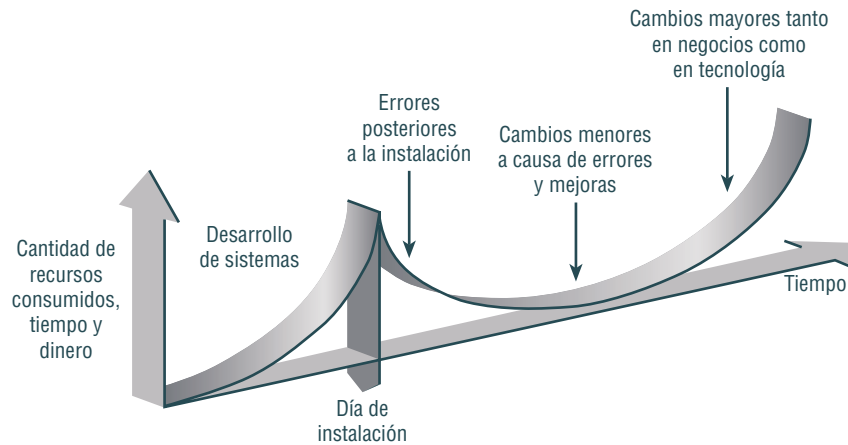


FIGURA 1.4

Algunos investigadores calculan que el tiempo invertido en el mantenimiento de sistemas podría llegar al 60 por ciento del tiempo total invertido en proyectos de sistemas.

FIGURA 1.5

Consumo de recursos durante la vida del sistema.



vos sistemas. Conforme se incrementa el número de programas escritos, también lo hace la cantidad de mantenimiento que requieren.

El mantenimiento se realiza por dos razones. La primera es la corrección de errores del software. No importa cuán exhaustivamente se pruebe el sistema, los errores se cuelan en los programas de cómputo. Los errores en el software comercial para PC se documentan como “anomalías conocidas”, y se corrigen en el lanzamiento de nuevas versiones del software o en revisiones intermedias. En el software hecho a la medida, los errores se deben corregir en el momento que se detectan.

La otra razón para el mantenimiento del sistema es la mejora de las capacidades del software en respuesta a las cambiantes necesidades de una organización, que por lo general tienen que ver con alguna de las siguientes tres situaciones:

1. Con frecuencia, después de familiarizarse con el sistema de cómputo y sus capacidades, los usuarios requieren características adicionales.
2. El negocio cambia con el tiempo.
3. El hardware y el software cambian a un ritmo acelerado.

La figura 1.5 ilustra la cantidad de recursos —por lo general tiempo y dinero— que se invierte en el desarrollo y mantenimiento de sistemas. El área bajo la curva representa la cantidad total invertida. Como puede apreciar, es probable que con el paso del tiempo el costo total del mantenimiento rebase el costo de desarrollar el sistema. Pasado un cierto tiempo es más factible realizar un nuevo estudio de sistemas, debido a que, evidentemente, el costo del mantenimiento continuo es mayor que el de la creación de un sistema de información completamente nuevo.

En síntesis, el mantenimiento es un proceso continuo durante el ciclo de vida de un sistema de información. Después de instalar el sistema de información, por lo general el mantenimiento consiste en corregir los errores de programación que previamente no se detectaron. Una vez corregidos estos errores, el sistema alcanza un estado estable en el cual ofrece un servicio confiable a sus usuarios. El mantenimiento durante este periodo podría consistir en eliminar algunos errores previamente no detectados y en actualizar el sistema con algunos cambios menores. Sin embargo, conforme pasa el tiempo y los negocios y la tecnología cambian, los esfuerzos de mantenimiento se incrementan de manera considerable.

USO DE HERRAMIENTAS CASE

A lo largo de este libro hacemos énfasis en la necesidad de un enfoque sistemático e integral para el análisis, diseño e implementación de sistemas de información. Reconocemos que para ser productivos, los analistas de sistemas deben realizar sus tareas de una manera organizada, precisa y minuciosa. Desde principios de la década de 1990, los analistas empezaron a beneficiarse de las herramientas de productividad, denominadas herramientas de Ingeniería de Software Asistida por Computadora (CASE, *Computer-Aided Software Engineering*), que

se crearon explícitamente para mejorar su trabajo rutinario mediante apoyo automatizado. De acuerdo con un estudio reciente, era más probable que los departamentos de sistemas de información con más de 10 empleados adoptaran las herramientas CASE que los departamentos con menos empleados. Los sistemas, procedimientos y prácticas administrativas de las organizaciones podrían restringir la difusión de las herramientas CASE. Los analistas de sistemas se apoyan en estas herramientas, desde el principio hasta el fin del ciclo de vida, para incrementar la productividad, comunicarse de manera más eficiente con los usuarios e integrar el trabajo que desempeñan en el sistema.

RAZONES PARA EL USO DE LAS HERRAMIENTAS CASE

Aumento en la productividad del analista Visible Analyst (VA) es una herramienta CASE que da al analista de sistemas la posibilidad de realizar planeación, análisis y diseño por medios gráficos, con el propósito de construir aplicaciones cliente-servidor y bases de datos complejas. Esta herramienta permite modelar los datos, procesos y objetos en diferentes formatos. Visible Analyst genera información sobre el modelo en muchas formas distintas, incluyendo COBOL, C, Visual Basic, SQL y XML. (En el sitio Web de este libro encontrará ejercicios de VA parcialmente terminados para las Experiencias con HyperCase y el Caso de la CPU que se sigue en los capítulos de este libro.)

Visible Analyst permite que sus usuarios dibujen y modifiquen diagramas con facilidad. De esta manera, el analista es más productivo tan sólo con la reducción del tiempo considerable que se invierte en dibujar y corregir manualmente diagramas de flujo de datos hasta que tengan una apariencia aceptable.

Un paquete de herramientas como Visible Analyst también mejora la productividad de grupos al dar a los analistas la posibilidad de compartir fácilmente el trabajo con otros miembros del equipo, quienes sólo tienen que abrir el archivo en sus PCs y revisar o modificar lo que se haya hecho. Esta facilidad de compartir el trabajo reduce el tiempo necesario para reproducir diagramas de flujo de datos y distribuirlos entre los miembros del equipo. Por tanto, en vez de requerir una distribución rigurosa y un calendario de respuestas con fines de retroalimentación, un paquete de herramientas permite a los miembros del equipo de análisis de sistemas trabajar con los diagramas siempre que lo necesiten.

Las herramientas CASE también facilitan la interacción entre miembros de un equipo al hacer que la diagramación sea un proceso iterativo y dinámico más que uno en el cual los cambios causen molestia y se conviertan en un freno para la productividad. En este caso la herramienta CASE para dibujar y grabar diagramas de flujo de datos ofrece un registro de la evolución de las ideas del equipo en lo concerniente a los flujos de datos.

Mejora de la comunicación analista-usuario Para que el sistema propuesto se concrete y sea útil en la práctica, es esencial una excelente comunicación entre analistas y usuarios durante todo el ciclo de vida del desarrollo de sistemas. El éxito de la futura implementación del sistema depende de la capacidad de analistas y usuarios para comunicarse de una manera eficiente. Hasta el momento, de las experiencias de analistas que utilizan herramientas CASE se desprende que su uso fomenta una mayor y más eficiente comunicación entre usuarios y analistas.

Analistas y usuarios por igual informan que las herramientas CASE ponen a su alcance un medio para comunicar aspectos del sistema durante su conceptualización. A través de apoyo automatizado que incluye salida en pantalla, los clientes pueden apreciar de inmediato cómo están representados los flujos de datos y otros conceptos del sistema, y pueden solicitar correcciones o cambios que hubieran tomado demasiado tiempo con herramientas anteriores.

El hecho de que un diagrama en particular sea considerado como útil por los usuarios o los analistas al final del proyecto es cuestionable. Lo importante es que este apoyo automatizado para muchas actividades de diseño del ciclo de vida es un medio para llegar a un fin al fungir como catalizador de la interacción analista-usuario. Los mismos argumentos que se utilizan para apoyar el rol de las herramientas CASE en el incremento de la productividad son igualmente válidos en este escenario; es decir, las tareas de dibujo, reproducción

y distribución toman mucho menos tiempo, de tal forma que es más sencillo compartir el trabajo en progreso con los demás usuarios.

Integración de las actividades del ciclo de vida La tercera razón para el uso de las herramientas CASE es integrar las actividades y proporcionar continuidad de una fase a la siguiente durante todo el ciclo de vida del desarrollo de sistemas.

Las herramientas CASE son especialmente útiles cuando una fase en particular del ciclo de vida requiere varias iteraciones de retroalimentación y modificaciones. Recuerde que la intervención de los usuarios puede ser importante en cada una de las fases. La integración de actividades mediante el uso subyacente de tecnologías facilita a los usuarios la comprensión de la manera en que se relacionan y dependen entre sí todas las fases del ciclo de vida.

Evaluar de manera precisa los cambios en el mantenimiento La cuarta, y probablemente una de las razones más importantes para el uso de herramientas CASE, es que permiten a los usuarios analizar y evaluar el impacto de los cambios en el mantenimiento. Por ejemplo, el tamaño de un elemento como un número de cliente podría requerir alargarse. La herramienta CASE pueden generar referencias cruzadas de cada pantalla, informe y archivo en el cual sea utilizado el elemento, dando lugar a un plan de mantenimiento integral.

HERRAMIENTAS CASE DE BAJO Y ALTO NIVEL

Las herramientas CASE se clasifican como de bajo nivel, de alto nivel e integradas, estas últimas combinando las de alto y bajo nivel en un solo conjunto. A pesar de que los expertos difieren en los criterios que definen con precisión cuáles son las herramientas CASE de alto nivel y cuáles las de bajo nivel, podría ser útil clasificarlas con base en los usuarios a los que dan apoyo. Las herramientas CASE de alto nivel ayudan principalmente a los analistas y diseñadores, en tanto que las de bajo nivel son utilizadas con más frecuencia por programadores y trabajadores que deben implementar los sistemas diseñados con herramientas CASE de alto nivel.

HERRAMIENTAS CASE DE ALTO NIVEL

Una herramienta CASE de alto nivel da al analista la posibilidad de crear y modificar el diseño del sistema. Toda la información relacionada con el proyecto se almacena en una enciclopedia denominada depósito CASE, una enorme colección de registros, elementos, diagramas, pantallas, informes e información diversa (véase la figura 1.6). Con la información del depósito se podrían generar informes que muestren dónde está incompleto el diseño o dónde contiene errores.

Las herramientas CASE de alto nivel también pueden apoyar la modelación de los requerimientos funcionales de una organización, ayudar a los analistas y usuarios a definir el alcance de un proyecto determinado y a visualizar la forma en que el proyecto se combina con otras partes de la organización. Además, algunas herramientas CASE de alto nivel pueden ayudar en la creación de prototipos de diseños de pantallas e informes.

HERRAMIENTAS CASE DE BAJO NIVEL

Las herramientas CASE de bajo nivel se utilizan para generar código fuente de computadora, eliminando así la necesidad de programar el sistema. La generación de código tiene varias ventajas:

1. El sistema se puede generar más rápido que si se tuviera que escribir todos los programas. No obstante, con frecuencia el periodo para familiarizarse con la metodología utilizada por el generador de código es muy largo, por lo que la generación del programa podría ser más lenta al principio. Además, es necesario ingresar por completo el diseño en el conjunto de herramientas, tarea que podría tomar un tiempo considerable.
2. La generación de código reduce el tiempo invertido en el mantenimiento. No hay necesidad de modificar, probar y depurar los programas de computadora. En lugar de eso, al

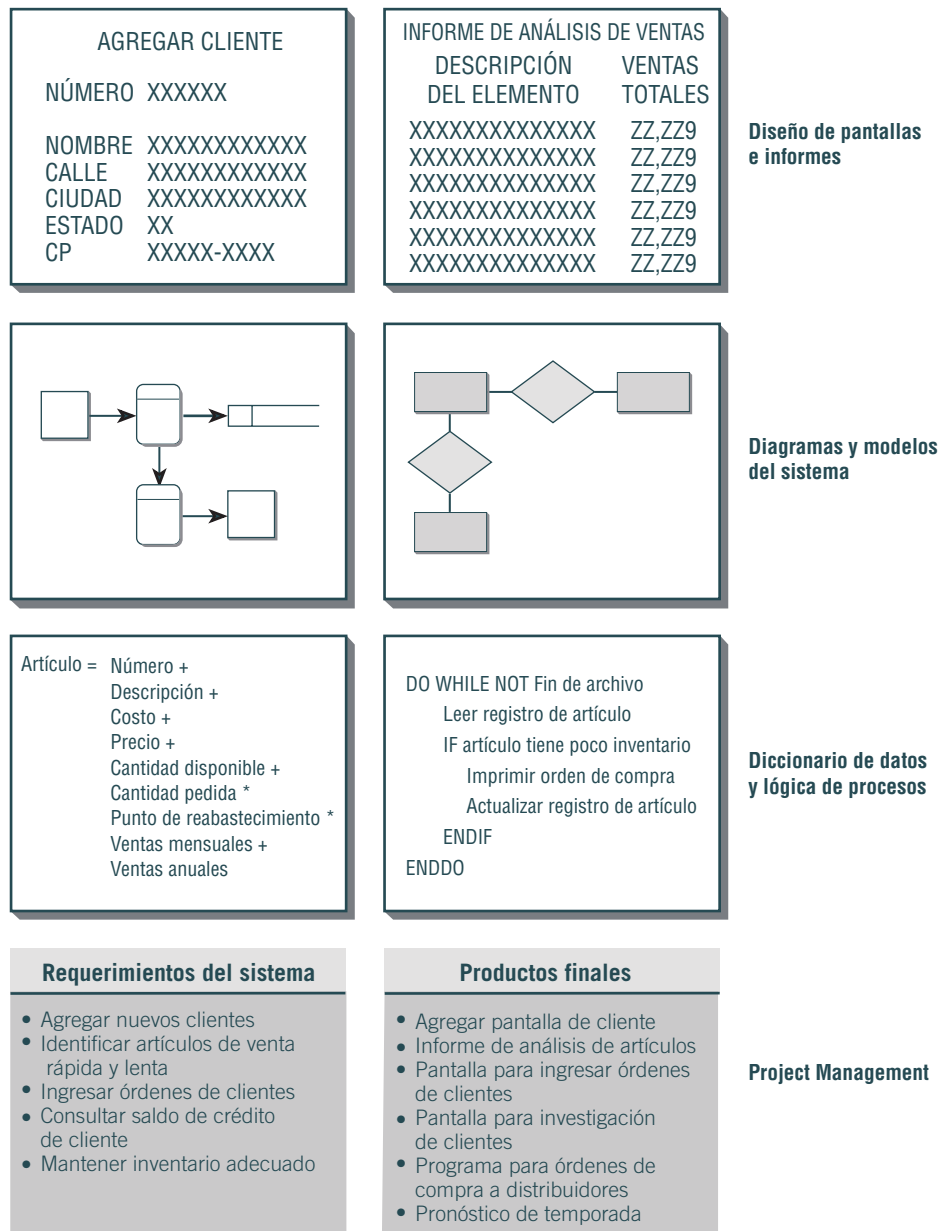


FIGURA 1.6
El concepto de depósito.

modificar el diseño CASE se vuelve a generar el código. Si se invierte menos tiempo en el mantenimiento, se tiene más tiempo para desarrollar nuevos sistemas y aligerar la acumulación de proyectos en espera de desarrollo.

- Más de un lenguaje de computadora, de tal manera que se facilita la migración de sistemas de una plataforma, digamos de *mainframe*, a otra, como una PC. Por ejemplo, la edición de VA para corporaciones puede generar código fuente en lenguajes de tercera generación como ANSI, COBOL o C.
- La generación de código ofrece una forma económica de ajustar los sistemas comerciales de fabricantes de sistemas a las necesidades de la organización. Con frecuencia, la modificación de esta clase de software implica un esfuerzo tan grande que su costo es mayor al de la compra del mismo. Con el software de generación de código, la compra de un diseño CASE y un depósito CASE para la aplicación permite al analista modificar el diseño y generar el sistema de cómputo modificado.
- El código generado está libre de errores de programación. Los únicos errores potenciales son los de diseño, los cuales se pueden minimizar produciendo informes de análisis CASE para garantizar que el diseño del sistema esté completo y correcto.

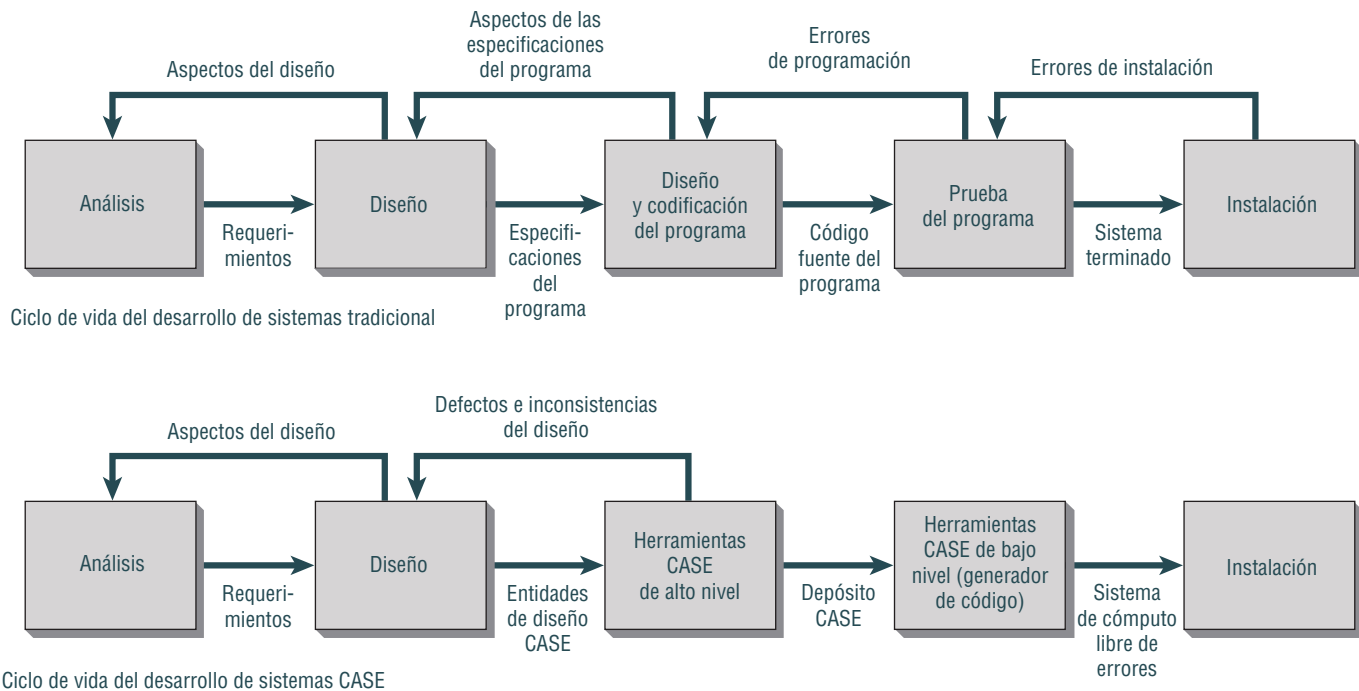


FIGURA 1.7

Ciclos de vida del desarrollo de sistemas CASE y tradicional.

La figura 1.7 ilustra los ciclos de vida del desarrollo de sistemas tradicional y CASE. Observe que las partes de codificación, prueba y depuración del programa se han eliminado en el ciclo de vida CASE.

INGENIERÍA INVERSA Y REINGENIERÍA DE SOFTWARE

La ingeniería inversa y la reingeniería de software son métodos para alargar la vida de programas anteriores, conocidos como software heredado. En ambos métodos se emplea software de reingeniería asistida por computadora (CARE, *Computer-Assisted Reengineering*) para analizar y reestructurar el código de computadora existente. En el mercado hay varios conjuntos de herramientas de ingeniería inversa.

Observe que el término *reingeniería* se utiliza en numerosos contextos diferentes de ingeniería, programación y negocios. Con frecuencia se emplea para denotar “reingeniería de procesos de negocios”, que es una forma de darle una nueva orientación a los procesos clave de una organización. Los analistas de sistemas pueden desempeñar un rol importante en la reingeniería de procesos de negocios, puesto que muchos de los cambios requeridos sólo se pueden lograr mediante el uso de tecnología de información novedosa.

La ingeniería inversa es lo opuesto a la generación de código. Como se ilustra en la figura 1.8, el código fuente de la computadora es examinado, analizado y convertido en entidades para el depósito. El primer paso de la ingeniería inversa de software es cargar, en el conjunto de herramientas, el código de programa existente (tal como se haya escrito en COBOL, C o cualquier otro lenguaje de alto nivel). Según el conjunto de herramientas de ingeniería inversa que se utilice, el código es analizado y las herramientas producen algunos o todos los elementos siguientes:

1. Estructuras de datos y elementos que describen los archivos y registros almacenados por el sistema.
2. Diseños de pantallas, si el programa es en línea.
3. Esquemas de informes para programas por lotes.
4. Un diagrama de estructura que muestra la jerarquía de los módulos del programa.
5. Diseño y relaciones de bases de datos.

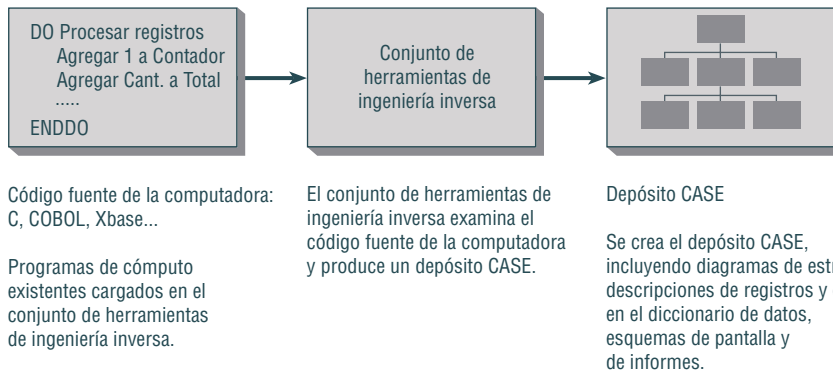


FIGURA 1.8
Conceptos de ingeniería inversa.

El diseño almacenado en el depósito podría modificarse o incorporarse en información de otro proyecto CASE. Cuando se terminan todas las modificaciones, el nuevo código del sistema puede volver a generarse. La reingeniería se refiere al proceso completo de convertir el código de programa al diseño CASE, modificar el diseño y volver a generar el nuevo código de programa.

Son varias las ventajas que se consiguen al utilizar un conjunto de herramientas de ingeniería inversa:

1. Reducción del tiempo requerido para el mantenimiento del sistema, con lo cual queda más tiempo para nuevos desarrollos.
2. Se genera documentación, que podría haber sido inexistente o mínima en los programas anteriores.
3. Se crean programas estructurados a partir de código de computadora no estructurado o pobremente estructurado.
4. Los cambios futuros al mantenimiento son más sencillos, porque se pueden realizar al nivel del diseño más que al nivel del código.
5. Es posible analizar el sistema con el fin de eliminar porciones sin utilizar de código de computadora, el cual aún podría estar presente en programas anteriores a pesar de que las revisiones hechas al programa a lo largo de los años lo hayan vuelto obsoleto.

ANÁLISIS Y DISEÑO DE SISTEMAS ORIENTADO A OBJETOS

El análisis y diseño orientado a objetos es un enfoque cuyo propósito es facilitar el desarrollo de sistemas que deben cambiar con rapidez en respuesta a entornos de negocios dinámicos. El capítulo 19 le ayuda a entender el análisis y diseño de sistemas orientados a objetos, en qué difiere del enfoque estructurado del SDLC y bajo qué circunstancias es apropiado utilizar un enfoque orientado a objetos.

Es difícil trabajar bien con técnicas orientadas a objetos en situaciones en las cuales sistemas de información complicados requieren mantenimiento, adaptación y rediseño de manera continua. Los enfoques orientados a objetos utilizan el estándar de la industria para la modelación de sistemas orientados a objetos, el lenguaje unificado de modelación (UML, *Unified Modeling Language*), para analizar un sistema en forma de modelo de casos de uso.

La programación orientada a objetos difiere de la programación tradicional de procedimientos en que la primera examina los objetos que conforman un sistema. Cada objeto es una representación en computadora de alguna cosa o suceso real. Los objetos pueden ser clientes, artículos, pedidos, etc. Los objetos se representan y agrupan en clases, que son óptimas para su reutilización y mantenimiento. Una clase define el conjunto de atributos y comportamientos que comparten los objetos que ésta contiene.

PROGRAMACIÓN EXTREMA Y OTRAS METODOLOGÍAS ALTERNAS

Aunque este libro se enfoca en la metodología que actualmente se utiliza de manera más amplia, en ocasiones el analista tendrá que reconocer que la organización se podría beneficiar de una metodología alterna. Quizá un proyecto de sistemas con un enfoque estructurado haya fallado, o quizá las subculturas que existen en la organización, compuestas por diferentes grupos de usuarios, parezcan más proclives a utilizar un método alterno. En un espacio tan breve no podríamos analizar adecuadamente estos métodos alternos, que merecen y han sido explicados en sus propios libros e investigaciones. Sin embargo, al mencionarlos aquí esperamos que usted tome conciencia de que, bajo ciertas circunstancias, su organización podría requerir una alternativa o complemento para un análisis y diseño estructurado y para el ciclo de vida del desarrollo de sistemas.

La programación extrema (XP, *Extreme Programming*) es un enfoque para el desarrollo de software que utiliza buenas prácticas de desarrollo y las lleva a los extremos. Se basa en valores, principios y prácticas esenciales. Los cuatro valores son la comunicación, la simplicidad, la retroalimentación y la valentía. Recomendamos a los analistas de sistemas que adopten estos valores en todos los proyectos que emprendan, no sólo cuando recurran a medidas de programación extrema.

Durante la fase de terminación de un proyecto, con frecuencia es necesario realizar ajustes en la administración del mismo. En el capítulo 3 veremos que XP puede garantizar la terminación exitosa de un proyecto ajustando recursos importantes como el tiempo, el costo, la calidad y el alcance. Cuando estas cuatro variables de control se incluyen adecuadamente en la planeación, se propicia un equilibrio entre los recursos y las actividades requeridas para completar el proyecto.

El llevar las prácticas de desarrollo al extremo es más recomendable cuando se siguen prácticas propias de XP. En el capítulo 6 describimos cuatro prácticas esenciales de XP: la liberación limitada, la semana de trabajo de 40 horas, alojar a un cliente en el sitio y el uso de la programación en parejas. A primera vista estas prácticas parecen extremas, pero como observará, podemos aprender algunas lecciones valiosas al incorporar muchos de estos valores y prácticas de XP en los proyectos de análisis y diseño de sistemas.

La creación de prototipos (que es diferente a la creación de prototipos que veremos en el capítulo 6) es uno de los métodos alternos más populares, junto con ETHICS, el enfoque de usar un campeón del proyecto, la Metodología Soft Systems y Multiview. La creación de prototipos, concebida originalmente en otras disciplinas y aplicada a los sistemas de información, surgió como respuesta a los extensos periodos de desarrollo asociados con el enfoque del ciclo de vida del desarrollo de sistemas y a la incertidumbre que existe con frecuencia en relación con los requerimientos de los usuarios. ETHICS, por su parte, se presentó como una metodología socio-técnica que combina soluciones sociales y técnicas. El enfoque de usar un campeón del proyecto, un concepto tomado de la mercadotecnia, adopta la estrategia de involucrar a una persona clave de cada área donde tiene influencia el sistema para garantizar el éxito del mismo. La Metodología Soft Systems fue concebida como una manera de modelar un mundo muchas veces caótico mediante el uso de “imágenes ricas”, ideogramas que captan los relatos característicos de una organización. Multiview se propuso como una forma de organizar y utilizar elementos de diversas metodologías en competencia.

RESUMEN

La información se puede considerar como un recurso organizacional. Como tal, se debe manejar con cuidado, al igual que los demás recursos. La disponibilidad de gran poder de cómputo en las organizaciones ha propiciado una explosión de información y, en consecuencia, se debe prestar mayor atención al manejo de la información generada.



“Bienvenido a Maple Ridge Engineering, al que en adelante llamaremos MRE. Esperamos que disfrute su trabajo de consultor de sistemas para nosotros. A pesar de que he trabajado aquí durante cinco años en diferentes actividades, recién fui asignado para fungir como apoyo administrativo para Snowden Evans, el jefe del nuevo Departamento de Capacitación y Administración de Sistemas. Ciertamente somos un grupo heterogéneo. Conforme se familiarice con la compañía, asegúrese de aprovechar todos sus conocimientos, tanto técnicos como relativos a las personas, para entender cómo somos e identificar los problemas y conflictos relacionados con nuestros sistemas de información que usted considere susceptibles de arreglar”.

“Para ponerlo al tanto, le diré que Maple Ridge Engineering es una compañía de ingeniería médica de mediano tamaño. El año pasado nuestros ingresos rebasaron 287 millones de dólares. Empleamos alrededor de 335 personas. Hay cerca de 150 empleados administrativos, así como personal directivo y de oficinas, como yo, y aproximadamente 75 profesionistas, como ingenieros, médicos y analistas de sistemas, y cerca de 110 empleados como dibujantes y técnicos”.

“Hay cuatro oficinas. A través de HyperCase, usted nos visitará en nuestras oficinas centrales en Maple Ridge, Tennessee. Tenemos otras tres sucursales al sur de Estados Unidos: Atlanta, Georgia; Charlotte, Carolina del Norte; y Nueva Orleans, Luisiana. Nos dará gusto que nos visite”.

“Por el momento, explore HyperCase con Netscape Navigator o Microsoft Internet Explorer”.

“Para aprender más sobre Maple Ridge Engineering como compañía, o para saber cómo entrevistar a nuestros empleados, quién utilizará los sistemas que usted diseñe y cómo son sus oficinas dentro de nuestra compañía, visite el sitio Web de este libro, y seleccione el vínculo **HyperCase**. Al desplegarse la pantalla de HyperCase, elija **Start** e ingresará a la recepción de Maple Ridge Engineering. A partir de aquí, puede empezar de inmediato su trabajo de consultoría”.

Este sitio Web contiene información útil para el proyecto y archivos que usted puede bajar a su computadora. Uno de los archivos contiene una serie de archivos de datos de Visible Analyst para utilizarse en HyperCase. Estos archivos incluyen una serie parcialmente construida de diagramas de flujo de datos, diagramas de entidad-relación y el depósito CASE. El sitio Web de HyperCase también contiene ejercicios adicionales. HyperCase está diseñado para facilitar su exploración, así que no ignore cualquier objeto o pista que encuentre en la página Web.

Los analistas de sistemas recomiendan, diseñan y dan mantenimiento a diversos tipos de sistemas, como los sistemas de procesamiento de transacciones (TPS), sistemas de automatización de la oficina (OAS), sistemas de trabajo del conocimiento (KWS) y sistemas de información gerencial (MIS). También crean sistemas orientados a la toma de decisiones, como los sistemas de apoyo a la toma de decisiones (DSS), sistemas expertos (ES), sistemas de apoyo a la toma de decisiones en grupo (GDSS), sistemas de trabajo colaborativo apoyados por computadora (CSCWS) y sistemas de apoyo a ejecutivos (ESS). Muchas aplicaciones se conciben originalmente para, o se migran a, la Web para apoyar el comercio electrónico.

El diseño y análisis de sistemas es un enfoque sistemático para identificar problemas, oportunidades y objetivos; para analizar los flujos de información de las organizaciones, y para diseñar sistemas de información computarizados destinados a solucionar problemas. Los analistas de sistemas se ven precisados a desempeñar diversos roles durante el transcurso de su trabajo. Algunos de estos roles son: (1) consultor externo para el negocio, (2) experto de apoyo técnico dentro de un negocio y (3) agente de cambio en situaciones tanto internas como externas.

Los analistas poseen un amplio rango de habilidades. En primer lugar, y más importante, el analista es un solucionador de problemas; alguien que disfruta el reto de analizar un problema e idear soluciones factibles. El analista de sistemas requiere habilidades de comunicación que le permitan relacionarse de manera significativa con diversas clases de gente diariamente, así como habilidades de computación. El involucramiento del usuario final es crítico para el éxito del proyecto.

Los analistas actúan de manera sistemática. El marco para este enfoque sistemático lo ofrece el ciclo de vida del desarrollo de sistemas (SDLC). Este ciclo de vida se puede dividir en siete fases secuenciales, aunque en realidad las fases se interrelacionan y con frecuencia se llevan a cabo de manera simultánea. Las siete fases son: identificación de problemas, oportunidades y objetivos; determinación de los requerimientos de información; análisis de las necesidades del sistema; diseño del sistema recomendado; desarrollo y documentación del software; prueba y mantenimiento del sistema, e implementación y evaluación del sistema.

Los paquetes de software automatizados, basados en PC, para el análisis y diseño de sistemas se conocen como herramientas de Ingeniería de Software Asistida por Computadora (CASE). Las cuatro razones para adoptar las herramientas CASE son: incrementar la productividad del analista, mejorar la comunicación entre analistas y usuarios, integrar las actividades del ciclo de vida, y analizar y valorar el impacto de los cambios en el mantenimiento. Los analistas también emplean enfoques de reingeniería asistida por computadora (CARE) para realizar ingeniería inversa de software y reingeniería con el propósito de extender la vida útil del software heredado.

El análisis orientado a objetos (OOA) y el diseño orientado a objetos (OOD) constituyen un enfoque distinto de desarrollo de sistemas. Estas técnicas se basan en los conceptos de la programación orientada a objetos, que han sido codificados en UML, un lenguaje estandarizado de modelación en el cual los objetos generados no sólo incluyen código referente a los datos sino también instrucciones acerca de las operaciones que se realizarán sobre los datos.

Cuando la situación particular de una organización así lo requiera, el analista podría dejar el SDLC y probar una metodología alterna. Un enfoque, denominado programación extrema (XP), lleva al límite las prácticas de análisis y diseño. La creación de prototipos, ETHICS, el uso de un campeón del proyecto, la Metodología Soft Systems y Multiview son enfoques de desarrollo que ofrecen perspectivas diferentes.

PALABRAS Y FRASES CLAVE

agente de cambio	experto en soporte técnico
análisis orientado a objetos (OOA)	groupware
análisis y diseño de sistemas	herramientas CASE
análisis y diseño de sistemas orientado a objetos (O-O)	información generada por computadora
analista de sistemas	Ingeniería de Software Asistida por Computadora (CASE)
aplicaciones de comercio electrónico	ingeniería inversa de software
asistente digital personal (PDA)	inteligencia artificial (AI)
CARE (reingeniería asistida por computadora)	lenguaje unificado de modelación (UML)
ciclo de vida del desarrollo de sistemas (SDLC)	mantenimiento de generación de código
comercio móvil	Metodología Soft Systems
consultor de sistemas	migración de sistemas
creación de prototipos	Multiview
depósito CASE	programación extrema (XP)
desarrollo rápido de aplicaciones (RDA)	reingeniería
diseño orientado a objetos (OOD)	sistemas de apoyo a ejecutivos (ESS)
enfoque de uso de un campeón del proyecto	sistemas de apoyo a la toma de decisiones (DSS)
ETHICS	sistemas de apoyo a la toma de decisiones en grupo (GDSS)

sistemas de automatización de la oficina (OAS)
sistemas de información gerencial (MIS)
sistemas de planeación de recursos empresariales (ERP)
sistemas de procesamiento de transacciones (TPS)

sistemas de trabajo colaborativo apoyado por computadora (CSCWS)
sistemas de trabajo del conocimiento (KWS)
sistemas expertos
sistemas Web
software de código abierto
software heredado

PREGUNTAS DE REPASO

1. Describa por qué es mejor considerar a la información como un recurso de la organización más que como un subproducto derivado de los negocios.
2. Defina lo que significa un sistema de procesamiento de transacciones.
3. Explique la diferencia entre los sistemas de automatización de la oficina (OAS) y los sistemas de trabajo del conocimiento (KWS).
4. Compare la definición de sistemas de información gerencial (MIS) y la de sistemas de apoyo a la toma de decisiones (DSS).
5. Defina el concepto *sistemas expertos*. ¿En qué son diferentes los sistemas expertos y los sistemas de apoyo a la toma de decisiones?
6. Enumere los problemas de interacción grupal que están destinados a resolver los sistemas de apoyo a la toma de decisiones en grupo (GDSS) y los sistemas de trabajo colaborativo apoyados por computadora (CSCWS).
7. ¿Cuál es el término más común, CSCWS o GDSS? Explique su respuesta.
8. Defina el concepto *comercio móvil* (*m-commerce*).
9. Enumere las ventajas de implementar aplicaciones en la Web.
10. ¿Cuál es la razón preponderante para diseñar sistemas ERP?
11. Defina el significado de software de código abierto.
12. Enumere las ventajas de utilizar técnicas de análisis y diseño de sistemas al desarrollar sistemas de información computarizados para negocios.
13. Mencione tres roles que debe desempeñar un analista de sistemas. Dé una definición de cada rol.
14. ¿Qué cualidades personales son de utilidad para el analista de sistemas? Enumérelas.
15. Mencione y describa brevemente las siete fases del ciclo de vida del desarrollo de sistemas (SDLC).
16. ¿En qué consiste el desarrollo rápido de aplicaciones (RAD)?
17. Defina ingeniería inversa de software y reingeniería en el contexto de reingeniería asistida por computadora (CARE).
18. Mencione las cuatro razones para adoptar herramientas CASE.
19. ¿Cuáles son los cuatro valores de la programación extrema?
20. Defina los conceptos *análisis orientado a objetos* y *diseño orientado a objetos*.
21. ¿Qué es el UML?

BIBLIOGRAFÍA SELECCIONADA

- Avison, D. E. y A. T. Wood-Harper, *Multiview: An Exploration in Information Systems Development*, Oxford: Blackwell Scientific Publications, 1990.
- Beath, C. M., "Supporting the Information Technology Champion", *MIS Quarterly*, vol. 15, núm. 3, septiembre de 1991, pp. 355-372.
- Checkland, P. B., "Soft Systems Methodology", *Human Systems Management*, vol. 8, núm. 4, 1989, pp. 271-289.
- Coad, P. y E. Yourdon, *Object-Oriented Analysis*, 2a. ed., Englewood Cliffs, NJ: Prentice Hall, 1991.
- Davis, G. B. y M. H. Olson, *Management Information Systems: Conceptual Foundation, Structure, and Development*, 2a. ed., Nueva York: McGraw-Hill, 1985.

- Holsapple, C. W. y A. B. Winston, *Business Expert Systems*, Homewood, IL: Irwin, 1987.
- Jackson, M. A., *Systems Development*, Englewood Cliffs, NJ: Prentice Hall, 1983.
- Kendall, K. E., "Behavioral Implications for Systems Analysis and Design: Prospects for the Nineties", *Journal of Management Systems*, vol. 3, núm. 1, 1991, pp. 1-4.
- Kendall, J. E. y K. E. Kendall, "Systems Analysis and Design", *In International Encyclopedia of Business and Management*, vol. 5, editado por M. Warner, pp. 4749-4759. Londres: Routledge, 1996.
- , "Information Delivery Systems: An Exploration of Web Push and Pull Technologies", *Communications of AIS*, vol. 1, artículo 14, 23 de abril de 1999.
- Laudon, K. C. y J. P. Laudon, *Management Information Systems*, 8a. ed., Upper Saddle River, NJ: Prentice Hall, 2004.
- Markoff, J., "Microsoft Sees Software 'Agent' as Way to Avoid Distractions", *New York Times*, 17 de julio de 2000, pp. C1, C4.
- Mumford, E. y M. Weir, *Computer Systems in Work Design—The ETHICS Method*, Londres: Associated Business Press, 1979.
- Sahami, M. S., Dumais, D. Heckerman y E. Horvitz, "A Bayesian Approach to Filtering Junk E-Mail", *AAAI Technical Report WS-98-05. AAAI Workshop on Learning for Text Categorization*, julio de 1998, Madison, WI.
- Sharma, S. y A. Rai, "CASE Deployment in IS Organizations", *Communications of the ACM*, vol. 43, núm. 1, enero de 2000, pp. 80-88.
- Stohr, E. A. y S. Viswanathan, "Recommendation Systems", en *Emerging Information Technologies: Improving Decisions, Cooperation, and Infrastructure*, editado por K. E. Kendall, pp. 21-44. Thousand Oaks, CA: Sage Publications, 1999.
- www.visible.com/products/index.html. Última visita el 27 de agosto de 2003.
- Yourdon, E., *Modern Structured Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1989.



ALLEN SCHMIDT, JULIE E. KENDALL Y KENNETH E. KENDALL

EMPIEZA EL CASO

1

Un día soleado y caluroso de fines de octubre, Chip Puller estacionó su automóvil y se encaminó a su oficina en la Central Pacific University. Era agradable la sensación de comenzar como analista de sistemas, y esperaba con ansia el momento de conocer a sus compañeros.

En la oficina, Anna Liszt se presenta a sí misma. “Nos han asignado para trabajar como equipo en un nuevo proyecto. Puedo ponerte al tanto de los detalles y después hacemos un recorrido por las instalaciones.”

“Me parece bien”, contesta Chip. “¿Cuánto tiempo llevas trabajando aquí?”

“Alrededor de cinco años”, responde Anna. “Empecé como analista programador, pero en los últimos años me he dedicado al análisis y el diseño. Espero que encontremos algunos métodos para incrementar nuestra productividad.”

“Háblame acerca del nuevo proyecto”, dice Chip.

“Bueno”, contesta Anna, “al igual que muchas organizaciones, tenemos un gran número de microcomputadoras con diferentes paquetes de software instalados. En 1980 había pocas microcomputadoras y una colección dispersa de software, pero se han incrementado rápidamente en los últimos años. El sistema actual utilizado para mantener el software y el hardware ha sido sobrepasado.”

“¿Qué hay de los usuarios? ¿A quién debo conocer? ¿Quién crees que sea importante que nos ayude con el nuevo sistema?”, pregunta Chip.

“Conocerás a todos, pero recientemente conocí a algunas personas importantes, y te diré lo que aprendí de ellas para que las recuerdes cuando te reúnas con ellas.”

“Dot Matricks es gerente de todos los sistemas de microcomputadoras de la Central Pacific. Al parecer podemos trabajar bien en conjunto. Ella es muy competente. Realmente le agrada mejorar la comunicación entre usuarios y analistas.”

“Será un placer conocerla”, especula Chip.

“Luego está Mike Crowe, el experto en mantenimiento de las microcomputadoras. Él es el más amable, pero está demasiado ocupado. Necesitamos ayudarlo a aligerar su carga. Cher Ware es compañera de Mike. Ella es muy liberal, pero no me malinterpretes, ella conoce su trabajo”, dice Anna.

“Debe ser divertido trabajar con ella”, comenta Chip.

“Es probable”, coincide Anna. “También conocerás a la analista financiera, Paige Prynter. Aún no la conozco.”

“Tal vez yo pueda ayudarte”, dice Chip.

“Por último, deberías —es decir, deberás— conocer a Hy Perteks, quien dirige bastante bien el Centro de Información. A él le encantaría que lográramos integrar las actividades de nuestro ciclo de vida.”

“Suena prometedor”, dice Chip. “Creo que la voy a pasar bien aquí.”

EJERCICIO

E-1. De la conversación preliminar que sostuvieron Chip y Anna, ¿en cuáles de los elementos que mencionaron se podrían aplicar las herramientas CASE?

