

# Como desenvolver sistemas de informação

## Capítulo

# 11

### OBJETIVOS DE ESTUDO

Ao terminar este capítulo, você será capaz de:

1. Identificar e descrever os principais passos de resolução de problemas para desenvolver novos sistemas de informação.
2. Avaliar modelos para mensurar o valor organizacional dos sistemas de informação.
3. Estimar os requisitos para gerenciar com sucesso a mudança criada pelos novos sistemas.
4. Avaliar os diversos métodos para desenvolver sistemas de informação.
5. Comparar as diversas metodologias para modelagem e projeto de sistemas.

### PLANO DO CAPÍTULO

Caso de abertura — *Um novo sistema de pedidos para os biscoitos das Girl Scouts*

11.1 Resolução de problemas e desenvolvimento de sistemas

11.2 Como compreender o valor empresarial dos sistemas e como gerenciar a mudança

11.3 Diferentes abordagens de desenvolvimento de sistemas

11.4 Como formatar e projetar sistemas: metodologias estruturadas e orientadas a objetos

Estudo de caso: resolvendo problemas organizacionais — *Como o sistema de escala de tripulação da Comair entrou em colapso*

### UM NOVO SISTEMA DE PEDIDOS PARA OS BISCOITOS DAS GIRLS SCOUTS

**Peanut Butter Patties, Caramel DeLites, Thin Mints:** os biscoitos fabricados pelas Girl Scouts (as 'bandeirantes' dos Estados Unidos) estão na preferência dos norte-americanos desde o primeiro *cookie drive* — um evento em que se vendem biscoitos em prol de uma causa benemerente — organizado pela entidade, em 1917. De lá para cá, as Girl Scouts se deram tão bem vendendo biscoitos que as vendas nessa área são a principal fonte de recursos da organização. As Girl Scouts vendem tantos biscoitos que coletar, contar e organizar a avalanche anual de pedidos se tornou um tremendo desafio.

O processo tradicional de pedido de biscoitos das Girl Scouts dependia de pilhas e pilhas de papel. Durante o pico de vendas, em janeiro, cada Girl Scout ia anotando suas vendas em um cartão de pedido individual e, ao completar o cartão, o entregava à líder da tropa. Em seguida, a líder passava as informações para um formulário de cinco vias e o entregava a uma voluntária da comunidade, que tabulava os pedidos. A partir daí, os pedidos eram repassados a uma sede regional do conselho, onde eram consolidados até formar os pedidos finais que seguiam para o fabricante, a ABC Cookies. Além de fazer pedidos, os voluntários e membros da tropa tinham de coordenar as entregas de biscoito — do fabricante para as seis regionais e de lá para os pontos de retirada, para cada bandeirante e finalmente para os consumidores propriamente ditos.

Apenas em 2004, o Patriots' Trail Girl Scout Council, que representa 65 comunidades e 18 mil Girl Scouts na área da grande Boston, vendeu mais de 1,6 milhão de caixas de oito variedades de biscoito. De acordo com sua diretora-executiva adjunta Deborah Deacetis, a papelada se tornara 'vassaladora'. "Os papéis trocavam de mãos muitas vezes. Havia inúmeras oportunidades para erros, pois a soma das colunas, a multiplicação dos preços por caixa e todos os cálculos tinham de ser feitos por diferentes pessoas, e em um prazo apertado."

O Patriots' Trail Girl Scout Council pensou primeiro em desenvolver um sistema computadorizado usando as ferramentas de desenvolvimento de aplicação e gerenciamento de banco de dados do Microsoft Access. Mas desenvolver essa alternativa lhes custaria 25 mil dólares, e levaria pelo menos de três a quatro meses até que o sistema estivesse pronto e funcionando. Era algo caro, demorado e complexo demais para as Girl Scouts. Além do software Microsoft Access, a entidade teria de comprar um servidor para rodar o sistema e ainda pagar os serviços de manutenção do site e da rede, para que o sistema ficasse disponível na Web.

Após ouvir a opinião dos consultores de gestão da Dovetail Associates, o conselho escolheu o QuickBase for Corporate Workgroups, da Intuit. O QuickBase é um serviço de software hospedado, baseado na Web, especial para pequenas empresas e grupos de trabalho corporativos. É particularmente adequado para desenvolver rapidamente aplicações de bancos de dados simples; além disso, não exige muito treinamento para ser usado. O QuickBase é customizável e projetado para coletar, organizar e compartilhar dados entre equipes situadas em diferentes lugares.

Em poucas horas, um consultor da Dovetail criou um protótipo QuickBase funcional com algumas funções básicas para as Girl Scouts. Usando esse software, o sistema inteiro foi desenvolvido, testado e implementado em apenas dois meses. O custo para desenvolver o sistema todo equivalia a apenas uma fração da solução Microsoft Access. Além disso, as Girl Scouts não precisam pagar por nenhum serviço de rede, software nem hardware, pois o QuickBase roda tudo para elas em seus próprios servidores. O serviço custa 500 dólares por mês para organizações com cem usuários, e 1.500 dólares por mês para organizações com mais de 500 usuários.

A solução QuickBase elimina a papelada e os erros de cálculo, pois oferece uma fonte de dados centralizada para todo o conselho, com uma fácil inserção on-line de pedidos de biscoito pela Web. Agora, as líderes de tropa coletam os cartões de pedidos das Girl Scouts e os inserem diretamente no sistema por meio de seus computadores residenciais conectados à Web. Com apenas alguns cliques de mouse, o conselho consolida os totais e transmite os pedidos eletronicamente para a ABC Cookies.

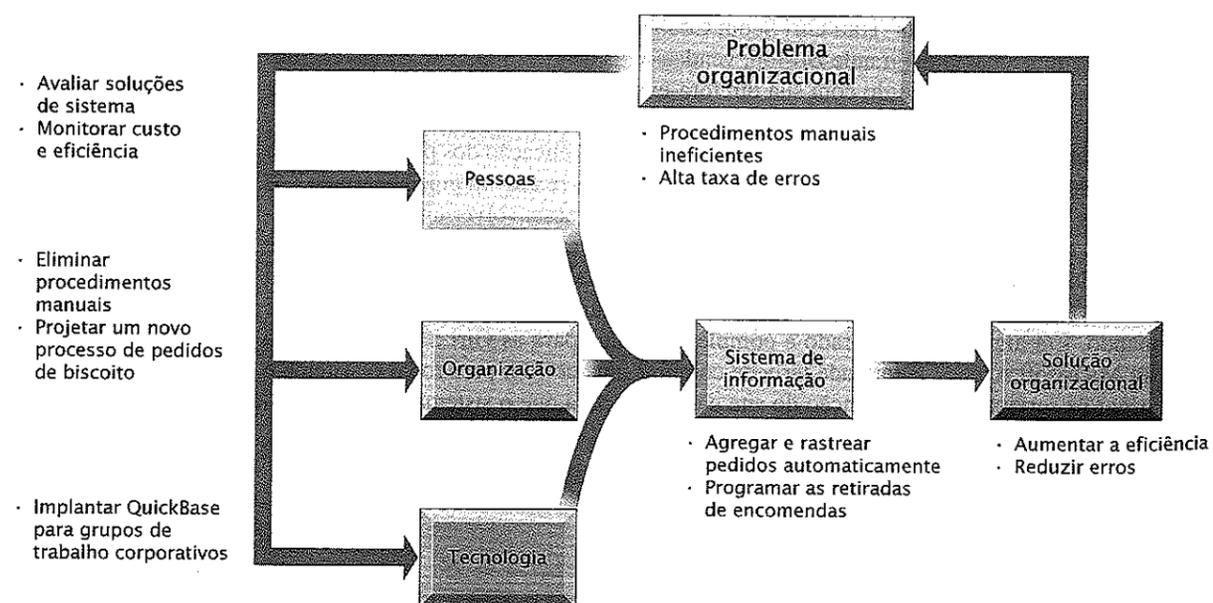
No passado, o conselho dependia de voluntários para administrar a papelada, deixando-a no escritório do conselho ou enviando-a pelo correio. "Agora, temos uma maneira efetiva de ver os pedidos que estão chegando", observa Deacetis. Assim que os pedidos são feitos, os líderes de seção locais podem controlar os dados em tempo real.

O Patriots' Trail Girl Scout Council também usa o sistema QuickBase para gerenciar o armazém Cookie Cupboard, onde os voluntários retiram suas encomendas de biscoito. Como os voluntários usam o sistema para fazer reservas, o armazém pode separar as encomendas com antecedência, economizando tempo e custos de gerenciamento de estoque. As transportadoras que entregam as encomendas de biscoito agora recebem instruções eletrônicas por meio do QuickBase, podendo criar assim programações de entrega eficientes.

Desde a sua implantação, o sistema QuickBase diminuiu a papelada em mais de 90 por cento, reduziu os erros a 1 por cento e cortou o tempo gasto pelos voluntários pela metade. O antigo sistema costumava levar dois meses para registrar todos os pedidos e determinar quais Girl Scouts seriam premiadas por vender mais biscoitos. Hoje, esse processo é feito em apenas 48 horas. ■

Fontes: "Girl Scouts unite behind order tracking", *Customer Relationship Management*, maio 2005; e Intuit Inc., "QuickBase customer stories: Patriots' Trail Girl Scouts and Dovetail Associates", [www.quickbase.com](http://www.quickbase.com), acessado em 11 jul. 2005.

A experiência do Patriots' Trail Girl Scout Council ilustra alguns passos necessários para projetar e desenvolver novos sistemas de informação. Ilustra também alguns benefícios que uma nova solução de sistema pode propor-



cionar. As Girl Scouts tinham um sistema manual ultrapassado, baseado em papel, para processar os pedidos de biscoito, o qual consumia um tempo excessivo e frequentemente levava a erros. A entidade experimentou várias soluções alternativas antes de optar por um novo sistema de pedidos usando o QuickBase como provedor de serviços de aplicação. Neste capítulo, examinamos a busca das Girl Scouts por uma solução de sistema e, ao mesmo tempo, descrevemos cada passo no desenvolvimento de um novo sistema de informação por meio do processo de resolução de problemas.

### ANTENA LIGADA

Ao longo de sua carreira, você certamente terá de trabalhar no desenvolvimento de um novo sistema para solucionar um desafio importante para sua empresa. No processo de desenvolvimento de novos sistemas, você deparará com muitas escolhas de hardware e software e terá de pensar em como reprojeter processos de negócios e cargos a fim de maximizar o valor extraído do novo sistema. Também precisará decidir se é melhor desenvolver sua própria solução de software ou comprar um pacote de software, ou ainda utilizar serviços Web on-line. Contudo, a decisão mais importante que você terá de tomar envolve compreender exatamente o que você quer que o software faça e qual valor ele trará à sua empresa. Este capítulo oferece a você uma metodologia para guiá-lo através do processo de resolução de problemas no desenvolvimento de novos sistemas de informação, bem como uma maneira de avaliar o valor empresarial dos novos sistemas.

- Se você está estudando economia e finanças ou contabilidade, no futuro ajudará os desenvolvedores de sistema a analisar os custos e benefícios de projetos de novos sistemas de informação e a avaliar sua viabilidade econômica.
- Se a sua área é recursos humanos, você analisará as mudanças nos fluxos de trabalho e nas responsabilidades de cada cargo resultantes de novos sistemas de informação; também providenciará o treinamento dos funcionários para os novos sistemas.
- Se você pretende atuar na gestão de operações, produção ou manufatura, trabalhará no desenvolvimento ou aprimoramento de sistemas de gerenciamento da cadeia de suprimentos e sistemas de planejamento dos recursos empresariais. Eles podem trazer benefícios importantes, mas estão entre os sistemas mais difíceis de implantar, pois exigem novas tecnologias e grandes mudanças na organização.
- Se você pretende trabalhar com vendas e marketing, utilizará novos sistemas baseados na Web para vendas on-line, ou um novo sistema de gerenciamento do relacionamento com o cliente, pois esses sistemas frequentemente recebem alta prioridade nos planos de sistemas de informação de muitas empresas.

Neste capítulo, você aprenderá a desenvolver novos sistemas de informação usando nosso modelo de resolução de problemas. Começamos descrevendo as atividades necessárias para analisar os problemas dos sistemas preexistentes e desenvolver novos sistemas. Em seguida, mostramos por que montar um sólido estudo de caso empresarial e gerenciar a mudança organizacional é tão importante para assegurar que o novo sistema de informação funcionará adequadamente e trará valor para a empresa. Finalmente, descrevemos diversas abordagens para o desenvolvimento de sistemas e diversas metodologias para modelá-los e projetá-los.

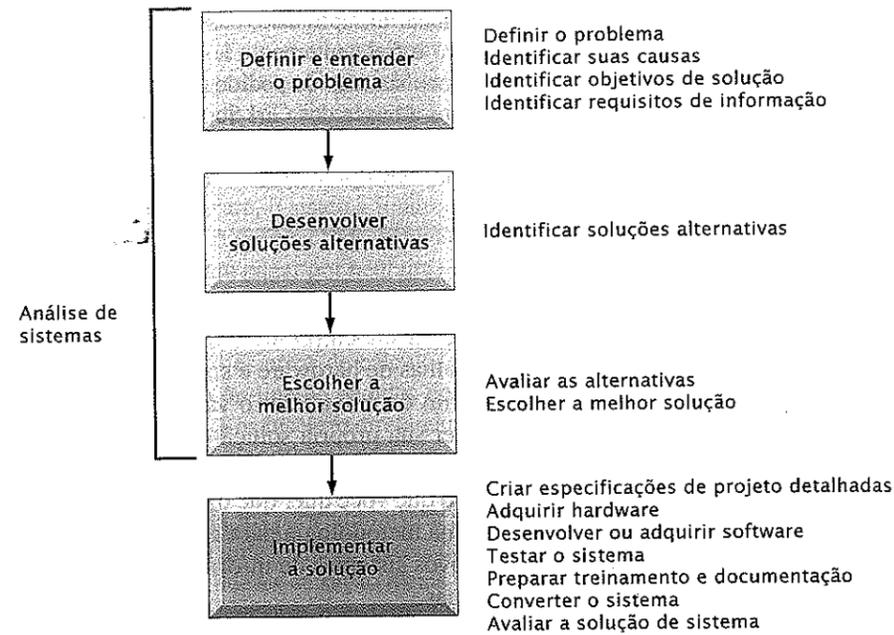
### 11.1 Resolução de problemas e desenvolvimento de sistemas

Já descrevemos o processo de resolução de problemas e como ele nos ajuda a analisar e compreender o papel dos sistemas de informação nas empresas. Esse processo de resolução de problemas é especialmente útil quando necessitamos desenvolver novos sistemas. Um novo sistema de informação é desenvolvido para solucionar um problema ou um conjunto de problemas que a organização reconhece estar enfrentando. Talvez os gerentes e funcionários achem que a empresa não está tendo um desempenho tão bom quanto o esperado, ou talvez percebam que a organização deveria tirar partido de novas oportunidades para ter um desempenho mais eficiente.

Vamos aplicar o processo de resolução de problemas ao desenvolvimento de sistemas. A Figura 11.1 ilustra os quatro passos que precisamos tomar: (1) definir e compreender o problema, (2) desenvolver soluções alternativas, (3) escolher a melhor solução e (4) implementar a solução.

**Figura 11.1**

Desenvolvimento de uma solução de sistema de informação.  
O desenvolvimento de uma solução de sistema de informação baseia-se no processo de resolução de problemas.



Um problema só pode ser solucionado se, antes, for adequadamente definido. Os membros da organização precisam estar de acordo que existe de fato um problema e que ele é sério. O problema precisa ser investigado para ser mais bem compreendido. Em seguida, vem um período em que se delineiam soluções alternativas, e depois outro em que se avalia cada alternativa e se seleciona a melhor solução. O estágio final é a implementação da solução, na qual um projeto detalhado para a solução é especificado, traduzido em um sistema físico, testado, introduzido na organização e, mais tarde, refinado conforme vai sendo usado ao longo do tempo.

No mundo dos sistemas de informação, temos um nome especial para essas atividades. A Figura 11.1 mostra que os primeiros três passos da resolução de problemas — identificar o problema, reunir informações, delinear soluções alternativas e tomar uma decisão sobre a melhor solução — são chamados de **análise de sistemas**.

### Definição e entendimento do problema

Definir o problema pode dar um certo trabalho, pois vários membros da empresa podem ter idéias diferentes sobre a natureza do problema e sua gravidade. O que causou o problema? Por que ele permaneceu? Por que não foi resolvido antes? Os analistas de sistemas normalmente reúnem fatos sobre os sistemas e problemas existentes examinando documentos, relatórios, procedimentos e operações de sistema e entrevistando usuários-chave.

No mundo dos negócios, os problemas de sistemas de informação geralmente resultam de uma combinação de fatores humanos, organizacionais e tecnológicos. Ao identificar um problema ou questão-chave, investigue de que tipo de problema se trata: é um problema humano, organizacional ou tecnológico, ou uma combinação das três coisas? Quais fatores humanos, organizacionais e tecnológicos contribuíram para o problema?

Uma vez que o problema tenha sido definido e analisado, é possível tomar algumas decisões sobre o que deve e pode ser feito. Quais são os objetivos de encontrar uma solução para o problema? O objetivo da empresa é reduzir custos, aumentar as vendas ou melhorar o relacionamento com clientes, fornecedores ou funcionários? Os gerentes dispõem de informações suficientes para a tomada de decisão? Qual informação é necessária para atingir esses objetivos?

No nível mais básico, os **requisitos de informação** de um novo sistema identificam quem precisa de qual informação, quando, onde e como. A análise de requisitos define cuidadosamente os objetivos do sistema modificado ou do novo sistema e desenvolve uma descrição detalhada das funções que o novo sistema deve desempenhar. Um sistema elaborado em torno do conjunto de requisitos errado ou terá de ser descartado, por causa de seu fraco desempenho, ou precisará sofrer grandes modificações. A Seção 11.3 descreve abordagens alternativas para identificação de requisitos que ajudem a minimizar esse problema.

Vamos voltar ao nosso caso de abertura sobre as Girl Scouts. O problema lá era que o processo de pedidos das Girl Scouts era quase totalmente manual e não conseguia comportar o grande número de voluntários e pedidos

de biscoito que precisavam ser coordenados. Em consequência, o processamento de pedidos era extremamente ineficiente, com altas taxas de erro, e os voluntários despendiam um tempo excessivo na organização de pedidos e entregas.

Do ponto de vista organizacional, as Girl Scouts são uma entidade sem fins lucrativos distribuída por uma grande área, tendo as vendas de biscoito como principal fonte de receita. As Girl Scouts dependem de voluntários com pouca ou nenhuma experiência em informática, negócios, vendas ou gerenciamento de pedidos e entregas. Elas não têm praticamente nenhuma fonte de financiamento, e os voluntários são pressionados pela falta de tempo. O processo de pedidos de biscoito das Girl Scouts exige vários passos e a coordenação de inúmeros grupos e organizações — as Girl Scouts em si, os voluntários, o escritório do conselho, a fábrica de biscoitos, as transportadoras e o armazém Cookie Cupboard.

O objetivo de uma solução para as Girl Scouts seria reduzir o tempo, o esforço e os erros no processo de pedido de biscoitos. Os requisitos de informação para a solução incluem a capacidade de rapidamente totalizar e organizar as transações de pedido para transmissão à ABC Cookies; a capacidade de rastrear pedidos por tipo de biscoito, tropa e Girl Scout; a capacidade de programar as entregas ao Cookie Cupboard; e a capacidade de programar retiradas de encomendas no Cookie Cupboard.

### Desenvolvimento de soluções alternativas

Quais soluções alternativas podem atingir esses objetivos e atender a esses requisitos de informação? A análise de sistemas traça os caminhos mais adequados a seguir dada a natureza do problema. Algumas soluções possíveis não exigem uma solução de sistemas de informação, mas sim um ajuste na administração, treinamento adicional ou refinamento dos procedimentos organizacionais existentes. Outras, porém, exigem modificações nos sistemas de informação existentes ou, ainda, um sistema de informação totalmente novo.

### Avaliação e escolha de soluções

A análise de sistemas inclui um **estudo de viabilidade** para determinar se cada solução proposta é viável, ou exequível, do ponto de vista financeiro, técnico e organizacional. O estudo de viabilidade determina se cada alternativa é um bom investimento, se a tecnologia necessária para o sistema está disponível e se poderia ser administrada pelos especialistas em sistemas de informação da empresa, e se a organização poderia administrar as mudanças introduzidas pelo sistema.

Um relatório escrito apresentando uma proposta de sistemas descreve os custos e benefícios, as vantagens e desvantagens de cada alternativa. Qual é a melhor solução do ponto de vista financeiro? Qual funciona melhor para a organização? A análise de sistemas detalha os custos e benefícios de cada alternativa e as mudanças que a organização deve fazer para usar a solução com eficiência. (Na seção seguinte, vemos em detalhes como determinar o valor empresarial dos sistemas e administrar a mudança.) Com base nesse relatório, a administração seleciona aquela que acredita ser a melhor solução para a empresa.

O Patriots' Trail Girl Scouts dispunha de três alternativas. A primeira era simplificar e unificar os processos existentes, continuando a depender de procedimentos manuais. No entanto, dado o grande número de Girl Scouts e pedidos de biscoito, assim como de relações com fabricantes e transportadoras, reprojeter e unificar o processo de entrega e pedidos manual não proporcionaria muitos benefícios. As Girl Scouts precisavam de uma solução automatizada que controlasse com precisão as milhares de transações de entrega e pedidos, reduzisse a papelada e criasse uma fonte de dados de vendas centralizada, em tempo real, que pudesse ser acessada pelo conselho e por voluntários individuais.

Uma segunda alternativa era desenvolver internamente um sistema de pedidos usando o Microsoft Access. Essa alternativa foi considerada muito cara, demorada e tecnicamente desafiadora para as Girl Scouts. Exigiria 25 mil dólares em custos iniciais de programação, além da compra de hardware e equipamento de rede para rodar o sistema e conectá-lo à Internet, assim como uma equipe especializada para operar e manter o sistema.

A terceira alternativa era criar rapidamente um sistema usando um provedor de serviço de aplicação (ASP). O QuickBase oferece modelos e ferramentas para a criação de sistemas de bancos de dados simples em espaços de tempo muito curtos, proporciona o hardware para rodar a aplicação, e o site pode ser acessado por inúmeros usuários pela Web. Essa solução não exigiria que as Girl Scouts comprassem nenhum hardware, software ou tecnologia de rede, tampouco mantivessem uma equipe especializada para apoiar o sistema. Por tudo isso, essa última alternativa foi a mais viável para as Girl Scouts.

## Implementação da solução

O primeiro passo na implementação de uma solução de sistema é criar especificações de projeto detalhadas. O projeto de sistemas mostra como a solução escolhida deve ser concretizada. O **projeto de um sistema** é a 'planta' ou modelo para uma solução de sistemas de informação e consiste em todas as especificações que executarão as funções identificadas durante a análise de sistemas. Essas especificações devem abordar todos os componentes organizacionais, tecnológicos e humanos da solução de sistema. A Tabela 11.1 lista os tipos de especificação que devem ser produzidos durante o projeto de sistemas.

Um consultor da Dovetail Associates reuniu os requisitos de informação e criou um projeto para o novo sistema de biscoitos das Girl Scouts. Estas seriam algumas das especificações de projeto para esse sistema:

Saída	Relatórios on-line Relatórios impressos Consultas on-line Transações de pedido para a ABC Cookies Boletos de entrega para a transportadora
Entrada	Formulário de entrada de dados do pedido Formulário de entrada de dados da tropa Formulário de entrada de dados da Girl Scout Formulário de entrada de dados de expedição/retirada
Interface de usuário	Interface gráfica na Web
Banco de dados	Banco de dados com arquivo de pedidos de biscoito, arquivo de entrega, arquivo de contato da tropa
Processamento	Calcular os totais dos pedidos por tipo de biscoito e número de caixas Rastrear pedidos por tropa e por Girl Scout Programar retiradas no Cookie Cupboard Atualizar os dados de endereço e filiação das tropas e das Girl Scouts
Procedimentos manuais	As Girl Scouts registram pedidos em formulários de papel As líderes de tropa coletam os cartões de pedido das Scouts e inserem os dados on-line
Segurança e controles	Senhas on-line Totais de controle
Conversão	Inserir dados da tropa e da Girl Scout Transferir dados da fábrica e da entrega Sistema de teste
Treinamento e documentação	Manual do sistema para os usuários Demonstração prática on-line Sessões de treinamento on-line Treinamento para as transportadoras e para a ABC Cookies quanto à aceitação automática de dados e instruções do sistema das Girl Scouts
Mudanças organizacionais	Projeto de cargos: os voluntários já não precisam tabular pedidos Projeto de processos: os pedidos são feitos manualmente em cartões, mas inseridos on-line no sistema Programar on-line retiradas de encomendas no Cookie Cupboard

## Finalização da implementação

Nos últimos estágios da implementação de uma solução de sistema, as seguintes atividades devem ser realizadas:

- **Seleção e aquisição de hardware.** Os desenvolvedores de sistema selecionam o hardware apropriado para a aplicação. Eles podem ou comprar o hardware de rede e os computadores necessários, ou alugá-los de uma provedora de tecnologia.

**Tabela 11.1**  
**Especificações de projeto de sistema**

Saída	Meio e conteúdo Sincronia
Entrada	Fluxo Entrada de dados
Interface de usuário	Feedback e administração de erros
Banco de dados	Modelo lógico de dados Requisitos de volume e velocidade Especificações de registros e arquivos
Processamento	Lógica de programação e cálculos
Procedimentos manuais	Quais atividades, quem as executa, quando, como e onde
Segurança e controles	Controles de acesso Controles de entrada, processamento e saída
Conversão	Método de teste Estratégia de conversão
Treinamento e documentação	Plataformas e módulos de treinamento Documentação de operações, usuários e sistemas
Mudanças organizacionais	Projeto dos processos Mudanças na estrutura organizacional

- **Desenvolvimento e programação de software.** O software pode ser programado internamente, sob medida, ou adquirido de uma fonte externa, como um fornecedor de outsourcing (terceirização), um fornecedor de pacotes de software aplicativo ou um provedor de serviços de aplicação (ASP).

As Girl Scouts decidiram não comprar nenhum software ou hardware adicional. O QuickBase oferece modelos para a geração de aplicações de bancos de dados simples. Os consultores da Dovetail usaram as ferramentas do QuickBase para criar rapidamente o software para o sistema. O sistema roda nos servidores QuickBase.

- **Testes.** O sistema é exaustivamente testado para se assegurar que está produzindo os resultados corretos. O processo de teste exige o teste detalhado de cada programa de computador, chamado **teste de unidade**, assim como o **teste de sistema**, que testa o funcionamento do sistema de informação como um todo. O **teste de aceite** provê a certificação final de que o sistema está pronto para ser usado em ambiente de produção. Testes de sistemas são avaliados por usuários e revisados pela administração. Quando todas as partes estiverem convencidas de que o novo sistema atende aos seus padrões, ele será formalmente aceito para instalação.

A equipe de desenvolvimento de sistemas trabalha com usuários para conceber um plano sistemático de teste. O **plano de teste** inclui todos os preparativos para a série de testes que acabamos de descrever. A Figura 11.2 mostra um exemplo de plano de teste que poderia ter sido usado para o sistema de biscoitos das Girl Scouts. A condição em teste é o acesso on-line ao registro preexistente de uma tropa específica.

- **Treinamento e documentação.** Usuários finais e especialistas em sistemas de informação precisam de treinamento para usar o novo sistema. Deve ser preparada uma detalhada **documentação** mostrando como o sistema funciona, tanto do ponto de vista técnico quanto do ponto de vista do usuário final.

O sistema de biscoitos das Girl Scouts oferece uma área on-line em que os usuários podem praticar a inserção de dados no sistema, seguindo instruções passo a passo. Também está disponível na Web um manual detalhado do sistema que pode ser baixado e impresso.

- **Conversão.** **Conversão** é o processo de passagem do sistema antigo para o novo. Há três principais estratégias de conversão: em paralelo, direta e da abordagem em fases.

Na **conversão em paralelo**, tanto o sistema antigo quanto o seu potencial substituto são executados juntos durante algum tempo, até que todos estejam seguros de que o novo funciona corretamente. O sistema antigo continua disponível como reserva, para o caso de problemas. A estratégia de **conversão direta** substitui completamente o sistema antigo pelo novo em uma data determinada, implicando o risco de que não há outro sistema de reserva ao qual recorrer caso surjam problemas. A **abordagem em fases** introduz o novo sistema em fases (por exemplo,

**Figura 11.2**

Exemplo de plano de teste para o sistema de biscoito das Girl Scouts. Ao desenvolver o plano de teste, é imperativo incluir as várias condições a serem testadas, os requisitos de cada condição testada e os resultados esperados. Planos de teste requerem entradas tanto de usuários finais quanto de especialistas em sistemas de informação.

Teste número: GS02-010	
Preparado por: A. Nelson	Data: 15 de fevereiro de 2006
Objetivo: Este subtteste verifica o acesso a um registro de tropa preexistente	
Ambiente específico: QuickBase for Workgroups	
<b>Descrição de procedimento:</b>	
Clique no link <b>Resumo da Minha Tropa</b> .	
Insira o Número de Tropa.	
<b>Resultado esperado:</b>	
Quando o usuário clicar em <b>Resumo da Minha Tropa</b> , a tela de <b>Resumo de Tropa</b> aparecerá.	
Quando o usuário inserir o Número de Tropa correto, o registro de Tropa aparecerá.	
Quando o usuário inserir o Número de Tropa incorreto, a mensagem de erro "Número de Tropa Incorreto" aparecerá.	
<b>Resultados do teste:</b>	
Tudo OK.	

primeiro os módulos para pedidos de biscoitos das Girl Scouts, depois os módulos para transmissão de pedidos e instruções à fábrica de biscoitos e às transportadoras).

- **Produção e manutenção.** Após a instalação do novo sistema e uma vez concluída a conversão, diz-se que o sistema está em **produção**. Durante esse estágio, ele será revisado por usuários e técnicos especializados para determinar em que medida atingiu seus objetivos originais e também para se decidir se há necessidade de revisões ou alterações. Mudanças em hardware, software, documentação ou procedimentos de um sistema em produção para corrigir erros, atender a novos requisitos ou melhorar a eficiência do processamento são chamadas de **manutenção**.

As Girl Scouts continuaram a melhorar e refinar seu sistema de biscoitos QuickBase. Em 2005, o sistema ficou mais eficiente para usuários com conexões lentas de Internet. Outros aperfeiçoamentos recentes incluem recursos para pagar os biscoitos mais rapidamente, inserir informações de tropa e pedidos iniciais sem ter de esperar por uma data de início específica e receber confirmação on-line de reservas para retirada de encomendas no Cookie Cupboard.

**Gerenciamento da mudança**

Desenvolver uma nova solução de sistema de informação não é apenas uma questão de instalar hardware e software. A empresa também precisa lidar com as mudanças organizacionais que a nova solução implicará — novas informações, novos processos de negócio e talvez novas relações hierárquicas e de poder para tomar decisões. Mesmo uma solução bem projetada pode não funcionar se não for introduzida na organização com muito cuidado. O processo de planejar a mudança em uma organização para que ela seja implantada de maneira ordenada e eficiente é tão decisivo para o sucesso ou fracasso das soluções de sistema de informação que dedicamos a seção seguinte a uma discussão detalhada sobre o assunto.

Para gerenciar a transição dos antigos processos manuais de pedido de biscoito para o novo sistema, as Girl Scouts teriam de informar as líderes de tropa e os voluntários sobre as mudanças nos procedimentos de pedido de biscoito, oferecer treinamento e providenciar recursos para responder às dúvidas que surgissem, à medida que pais e voluntários começassem a usar o sistema. Precisaríamos também trabalhar com a ABC Cookies e as transportadoras para estabelecer os novos procedimentos de transmissão e entrega de pedidos.

**11.2 Como compreender o valor empresarial dos sistemas e como gerenciar a mudança**

Sua empresa desenvolveu o que parece ser uma excelente solução de sistema. Mas, na prática, o sistema não funciona adequadamente, ou não produz os benefícios esperados. Se isso estiver acontecendo, sua empresa não está sozinha. Existe uma altíssima taxa de fracasso entre os projetos de sistemas de informação, ou porque as organizações avaliaram incorretamente o valor empresarial deles, ou porque não conseguiram administrar a mudança orga-

nizacional exigida pela nova tecnologia. Vamos analisar com mais cuidado o que precisa ser feito para assegurar que a solução de informação seja bem-sucedida.

**Como montar o estudo de caso empresarial para o novo sistema**

Conforme ressaltamos anteriormente, a análise de sistemas inclui uma avaliação da viabilidade econômica de cada solução — ou seja, se cada uma das soluções alternativas representa um bom investimento para a empresa. Não será uma boa prática empresarial gastar dinheiro em um sistema de informação a menos que seus benefícios superem seus custos. Você precisa montar um sólido estudo de caso empresarial para a solução escolhida.

**Aspectos financeiros**

A Tabela 11.2 lista alguns dos custos e benefícios mais comuns dos sistemas. Os **benefícios tangíveis** podem ser quantificados e pode-se atribuir a eles um valor monetário. Os **benefícios intangíveis** — como serviço de atendimento ao cliente mais eficiente ou processo de decisão aperfeiçoado —, por sua vez, não podem ser quantificados imediatamente. Contudo, mesmo os sistemas que produzem benefícios preponderantemente intangíveis podem tornar-se bons investimentos caso gerem ganhos quantificáveis no longo prazo.

Para determinar os benefícios de uma solução específica, você precisa calcular seus custos totais e seus benefícios totais. Obviamente, uma solução cujos custos superem os benefícios deve ser rejeitada. No entanto, mesmo que os benefícios superem os custos, ainda é necessária uma análise financeira mais detalhada para determinar se o investimento representa um bom retorno sobre o capital investido.

Em geral, usam-se métodos de orçamento de capital, como valor presente líquido, taxa interna de retorno (TIR) ou taxa de retorno contábil sobre o investimento (ROI) para avaliar a solução de sistema de informação proposta como um investimento.

Alguns dos benefícios tangíveis obtidos pelas Girl Scouts foram maior produtividade e custos operacionais mais baixos resultantes da automatização do processo de pedido e da redução de erros. Os benefícios intangíveis incluem a maior satisfação no trabalho voluntário e o aperfeiçoamento das operações.

**Aspectos não financeiros**

Os sistemas de informação também criam valor por fortalecer estrategicamente a empresa — ao, por exemplo, diferenciar produtos e serviços, fortalecer os laços com clientes e fornecedores, ou proporcionar flexibilidade para

<b>Custos de implantação</b>	<b>Benefícios intangíveis</b>
Hardware	Melhor utilização de ativos
Telecomunicações	Controle de recursos melhorado
Software	Planejamento organizacional melhorado
Pessoal	Maior flexibilidade organizacional
	Informação mais oportuna
<b>Custos operacionais</b>	Mais informação
Tempo de processamento computacional	Aprendizagem organizacional melhorada
Manutenção	Cumprimento de exigências legais
Equipe de operação	Aumento da boa vontade dos funcionários
Tempo do usuário	Aumento da satisfação com o trabalho
Custos de formação continuada	Processo de decisão melhorado
Custos de infra-estrutura	Operações melhoradas
<b>Benefícios tangíveis</b>	Maior satisfação do cliente
Produtividade aumentada	Melhor imagem corporativa
Custos operacionais mais baixos	
Força de trabalho reduzida	
Despesas mais baixas com computadores	
Custos mais baixos de fornecedores externos	
Custos burocráticos e profissionais mais baixos	
Taxa reduzida de crescimento de despesas	
Custos de infra-estrutura reduzidos	
Vendas aumentadas	

**Tabela 11.2**

**Custos e benefícios dos sistemas de informação**

responder rapidamente a mudanças futuras. O estudo de caso empresarial para uma solução pode abordar esses aspectos estratégicos de longo prazo.

Por exemplo, é importante ter certeza de que a solução escolhida se encaixa na estratégia e no plano de negócio geral da empresa. Grandes empresas muitas vezes criam um **plano de sistemas de informação** para essa finalidade. A Tabela 11.3 ilustra os principais componentes de um plano desse tipo.

**Tabela 11.3****Plano de sistemas de informação**

1. Propósito do plano
  - Visão geral do conteúdo do plano
  - Organização atual e futura da empresa
  - Processos de negócios-chave
  - Estratégia de gerenciamento
2. Plano de negócios estratégico
  - Situação atual
  - Organização empresarial atual
  - Ambientes mutáveis
  - Principais metas do plano de negócios
  - Plano estratégico da empresa
3. Sistemas atuais
  - Principais sistemas de apoio a funções e processos
  - Recursos atuais de infra-estrutura
    - Hardware
    - Software
    - Banco de dados
    - Telecomunicações e Internet
  - Dificuldades de atender a requisitos empresariais
  - Demandas futuras previstas
4. Novas tendências
  - Novos projetos de sistema
    - Descrições de projeto
    - Princípios racionais da empresa
    - Papel estratégico das aplicações
  - Novos recursos de infra-estrutura requeridos
    - Hardware
    - Software
    - Banco de dados
    - Telecomunicações e Internet
5. Estratégia de gerenciamento
  - Planos de aquisição
  - Marcos e *timing*
  - Realinhamento organizacional
  - Reorganização interna
  - Controles de gerenciamento
  - Principais iniciativas de treinamento
  - Estratégia de pessoal
6. Plano de implementação
  - Dificuldades previstas na implementação
  - Relatórios de progresso
7. Requisitos orçamentários
  - Requisitos
  - Economias potenciais
  - Financiamento
  - Ciclo de aquisição

O plano contém uma declaração de metas corporativas e especifica como a tecnologia de informação apóia a realização dessas metas. O relatório mostra como as metas gerais serão alcançadas pelos projetos de sistemas específicos. Estabelece datas e os marcos específicos buscados que podem ser usados mais tarde para avaliar o progresso do plano em termos de quantos objetivos foram realmente atingidos dentro do cronograma especificado. O plano indica as decisões-chave da administração concernentes à aquisição de hardware; telecomunicações; centralização/descentralização de autoridade, dados, hardware e mudanças organizacionais requeridas.

O plano também precisa descrever as mudanças organizacionais, incluindo requisitos de gerenciamento e treinamento de funcionários; mudanças nos processos de negócios; e mudanças em autoridade, estrutura ou prática de gerenciamento. Ao montar o estudo de caso para um novo projeto de sistemas de informação, você deve mostrar como o sistema proposto se encaixa nesse plano.

Uma vez que você tenha determinado o rumo geral do desenvolvimento de sistemas, a **análise de carteira (portfólio)** ajudará você a avaliar alternativas de projetos de sistemas. A análise de carteira faz um inventário de todos os ativos e projetos de sistemas de informação da empresa, incluindo infra-estrutura, contratos de terceirização e licenças. Assim como uma carteira financeira, essa carteira de investimentos em sistemas de informação pode ser descrita como tendo um certo perfil de risco e benefício para a empresa (veja a Figura 11.3). Cada projeto de sistema de informação tem seu próprio conjunto de riscos e benefícios. As empresas tentam aumentar o retorno sobre suas carteiras de sistemas de informação contrabalancando o risco e o retorno de todos os seus investimentos em sistemas.

Obviamente, você deve começar focalizando sistemas de alto benefício e baixo risco — isto é, os que prometem retornos rápidos e seguros. Em segundo lugar, devem ser examinados os sistemas de alto benefício e alto risco; sistemas de baixo benefício e alto risco devem ser totalmente evitados, ao passo que os de baixo benefício e baixo risco devem ser reexaminados, a fim de verificar se existe a possibilidade de remodelá-los ou substituí-los por sistemas mais convenientes, com benefícios mais altos. Graças à análise de carteira, a administração pode determinar a combinação ideal de investimento de retorno para as empresas, contrabalancando projetos mais arriscados e de alto retorno com projetos mais seguros, porém menos rentáveis.

O ABN Amro Bank NV, de Amsterdã, usa a análise de carteira para avaliar e classificar as centenas de projetos de sistemas de informação nos 55 países onde opera. A análise de carteira ajudou o banco a estabelecer prioridades entre as demandas conflitantes de suas unidades de negócios, e a justificar as decisões da administração quanto a quais sistemas desenvolver (Hoffman, 2005).

Outro método para chegar a uma decisão quanto a alternativas de sistemas é o **modelo de pontuação (scoring)**. Nesse método, atribui-se às alternativas uma pontuação única, baseada no grau de atendimento a objetivos selecionados. A Tabela 11.4 mostra parte de um modelo de pontuação simples que poderia ter sido usado pelas Girl Scouts para avaliar suas alternativas de sistema. A primeira coluna apresenta os critérios que os tomadores de decisão usam para avaliar os sistemas. Essa tabela mostra que as Girl Scouts atribuíam grande importância aos recursos de processamento de pedidos, à facilidade de uso, à capacidade de atender usuários em lugares diferentes e ao baixo custo. A segunda coluna apresenta os pesos que os tomadores de decisão atribuem aos critérios. As colunas 3 e 5 mostram a porcentagem de requisitos para cada função que a alternativa atende. A pontuação de cada alternativa é calculada multiplicando-se a porcentagem de requisitos atendidos para cada função pelo peso atribuído a essa função. A solução QuickBase tem a maior pontuação total.

		Risco do projeto	
		Alto	Baixo
Benefícios potenciais para a empresa	Alto	Examinar prudentemente	Identificar e desenvolver
	Baixo	Evitar	Projetos de rotina

**Figura 11.3**

Uma carteira de sistemas. As empresas devem examinar sua carteira de projetos em termos de benefícios potenciais e riscos prováveis. Certos tipos de projeto devem ser totalmente evitados e outros, desenvolvidos rapidamente. Não existe a fórmula ideal, já que empresas de diferentes setores têm perfis diferentes.

**Tabela 11.4****Exemplo de um modelo de pontuação para o sistema de biscoitos das Girl Scouts**

Critério	Pesos	Sistema Microsoft Access (%)	Pontuação do sistema Microsoft Access	Sistema Quickbase (%)	Pontuação do sistema Quickbase
<b>1.0 Processamento de pedidos</b>					
1.1 Inserção de pedidos on-line	5	67	335	83	415
1.2 Rastreamento de pedidos por tropa	5	81	405	87	435
1.3 Rastreamento de pedido por Girl Scout	5	72	360	80	400
1.4 Reserva de retiradas no armazém	3	66	198	79	237
Processamento de pedido total			1.298		1.487
<b>2.0 Facilidade de uso</b>					
2.1 Acesso Web a partir de diferentes lugares	5	55	275	92	460
2.2 Pouco tempo de treinamento	4	79	316	85	340
2.3 Telas amigáveis e formulários para inserção de dados	4	65	260	87	348
Facilidade de uso total			851		1.148
<b>3.0 Custos</b>					
3.1 Custos de software	3	51	153	65	195
3.2 Custos de hardware (servidor)	4	57	228	90	360
3.3 Custos de suporte e manutenção	4	42	168	89	356
Custos totais			549		911
Pontuação total			2.698		3.546

### Novos desafios de sistema

Uma porcentagem muito grande de sistemas de informação fracassa em trazer benefícios ou resolver os problemas que deveria resolver porque o processo de mudança organizacional que cercou sua implantação não foi adequadamente abordado. Talvez o projeto do sistema em si não consiga captar as necessidades essenciais da empresa nem melhorar seu desempenho. Talvez a informação não seja fornecida com rapidez suficiente para ser útil; ou talvez esteja em um formato impossível de assimilar e usar; ou, ainda, retrate partes da informação que não interessam.

O sistema pode ter sido projetado com uma interface do usuário deficiente. A **interface do usuário** é a parte do sistema com a qual os usuários finais interagem. Por exemplo, um formulário de entrada ou uma tela de entrada de dados on-line podem estar tão mal estruturados que ninguém queira apresentar seus dados. Certos sites desencorajam o visitante a prosseguir na sua exploração: as páginas estão saturadas e mal estruturadas, ou não se consegue encontrar rapidamente o que se procura, ou ainda se perde muito tempo para acessar e exibir a página.

Alguns sistemas operam sem nenhuma dificuldade significativa, mas seu custo de implementação e execução em regime de produção pode ficar muito acima do orçamento. Nesse caso, o valor empresarial da informação que eles fornecem não justifica as despesas excessivas.

Para gerenciar efetivamente a mudança organizacional que cerca a introdução de um novo sistema de informação, você deve examinar o processo de implementação. Uma definição mais ampla de **implementação** engloba todas as atividades organizacionais desenvolvidas em prol da adoção e do gerenciamento de uma inovação tal como um novo sistema de informação. No processo de implementação, o analista de sistema é um **agente de mudança**. Ele não somente desenvolve soluções técnicas, como também redefine as interações, atividades de serviço e rela-

ções de poder entre os vários grupos organizacionais. Catalisador de todo o processo de mudança, o analista é responsável por assegurar que as alterações criadas por um novo sistema sejam aceitas por todas as partes envolvidas.

O sucesso ou fracasso de um novo sistema de informação depende, em grande medida, do papel dos usuários, do grau de respaldo da administração, do nível de risco e complexidade do projeto de implementação e de como o processo de implementação em si é gerenciado.

### Envolvimento e influência do usuário

Se os usuários estiverem profundamente envolvidos no desenvolvimento do sistema, terão mais oportunidades de moldá-lo de acordo com suas prioridades e necessidades profissionais, e mais oportunidades para controlar o resultado. Também tenderão a reagir positivamente quando o sistema estiver finalizado, pois terão atuado como participantes ativos no processo de mudança. Incorporar o conhecimento e a perícia técnica dos usuários gera soluções melhores.

O relacionamento entre usuários finais e especialistas em sistemas de informação tem sido, tradicionalmente, uma área problemática para os esforços de implementação de sistemas de informação, pois esses dois grupos tendem a ter diferentes formações técnicas e pessoais, interesses e prioridades. Essas diferenças criam a **dificuldade de comunicação usuário-projetista**. Especialistas em sistemas de informação frequentemente têm uma orientação altamente técnica à resolução do problema. Buscam soluções técnicas nas quais as eficiências de hardware e software são otimizadas, em detrimento da facilidade de utilização ou da efetividade organizacional. Os usuários, por sua vez, preferem sistemas orientados à resolução de problemas empresariais ou que facilitem as tarefas organizacionais. Frequentemente as orientações dos dois grupos são tão conflitantes que eles parecem estar falando idiomas diferentes.

Essas diferenças estão ilustradas na Tabela 11.5, que descreve as preocupações típicas de usuários finais e especialistas técnicos (projetistas de SI) em relação ao desenvolvimento de um novo sistema de informação. Problemas de comunicação entre usuários finais e projetistas são a maior razão pela qual os requisitos de usuários não são adequadamente incorporados aos sistemas, e os usuários acabam afastados do processo de implementação.

### Apoio e comprometimento da organização

Quando um projeto de sistema de informação tem o respaldo e o comprometimento dos vários níveis da administração, é mais provável que receba prioridade mais alta tanto dos usuários como do pessoal de TI. O respaldo da administração também garante que um projeto de sistema receberá financiamento e recursos suficientes para ser bem-sucedido. Na verdade, todas as mudanças em hábitos de trabalho e procedimentos e quaisquer realinhamentos organizacionais associados com um novo sistema dependem do apoio efetivo da administração.

### Nível de complexidade e risco

Alguns projetos de desenvolvimento de sistemas têm mais probabilidade de fracassar ou sofrer atrasos porque seu nível de risco é muito alto. Projetos grandes e complexos — o que se mede pelo dinheiro gasto, pela quantidade de mão-de-obra envolvida na implementação, pelo tempo consumido e pelo número de unidades organizacionais afetadas — implicam um risco muito mais alto. Projetos de sistemas de grande escala, incluindo implementações de sistemas de planejamento de recursos empresariais ou de gerenciamento do relacionamento com os clientes, têm índices de fracasso de 50 a 75 por cento mais altos que os outros, pois são muito complexos e difíceis de controlar (Concours Group, 2000; Xia e Lee, 2004). Os riscos também são mais altos no caso de sistemas em que os requi-

Preocupações do usuário	Preocupações do projetista
O sistema fornecerá a informação de que necessito para meu trabalho?	Quanto espaço de disco de armazenagem o arquivo-mestre ocupa?
Com que rapidez posso acessar os dados?	Quantas linhas de programa serão necessárias para executar essa função?
Com que facilidade posso recuperá-los?	Como podemos reduzir tempo de CPU ao rodar o sistema?
Para inserir dados no sistema, quanta burocracia precisarei enfrentar?	Qual é a maneira mais eficiente de armazenar esses dados?
Como o sistema vai se ajustar à minha programação diária?	Que sistema de gerenciamento de banco de dados devemos usar?

**Tabela 11.5****A dificuldade de comunicação usuário-projetista**

sitos de informação não estejam claros e bem delineados, ou em que a equipe de projeto precise lidar com uma tecnologia nova e complexa.

**Qualidade do gerenciamento de projetos**

Um projeto de desenvolvimento de sistemas precisa ser cuidadosamente gerenciado para que se tenha certeza de que as tarefas serão finalizadas a tempo e que todos os grupos envolvidos no novo sistema trabalharão juntos com eficiência. Muitas vezes são esquecidos elementos básicos do sucesso, como o treinamento necessário para que os usuários finais se sintam confortáveis com o novo sistema. Sem o gerenciamento adequado, um projeto de desenvolvimento de sistemas leva mais tempo para ser finalizado e, com frequência, excede o orçamento designado. Para completar, o sistema de informação resultante tem grande probabilidade de ser tecnicamente inferior e não conseguir trazer nenhum benefício à organização.

**Como gerenciar a mudança efetivamente**

Se esses problemas parecem desanimadores, não desista! Existem estratégias que você pode seguir para aumentar a chance de sucesso de uma solução de sistema. Se o novo sistema envolve tecnologia complexa e desafiadora, você pode recrutar líderes de projeto com sólida experiência técnica e administrativa. A terceirização ou o uso de consultores externos são opções a considerar caso sua empresa não tenha pessoal com a perícia e os conhecimentos técnicos necessários. **Ferramentas formais de planejamento e de controle**, tais como o Pert (*Program Evaluation and Review Technique*) ou os gráficos de Gantt (veja a Figura 11.4), auxiliam o gerenciamento de projetos, pois enumeram as atividades específicas que formarão o projeto, sua duração e a seqüência e prazo de cada tarefa.

Você pode superar a resistência do usuário promovendo sua participação (tanto para estimular seu comprometimento quanto para melhorar o projeto), tornando a formação e o treinamento do usuário facilmente disponíveis e proporcionando mais incentivos para que os usuários cooperem. Os usuários finais podem transformar-se em membros ativos da equipe de projetos, assumir papéis de liderança e encarregar-se da instalação do sistema e do treinamento.

Você precisa prestar atenção especial às áreas em que os usuários fazem interface com o sistema, usando sensibilidade quanto às questões ergonômicas. **Ergonomia** refere-se à interação entre pessoas e máquinas no ambiente de trabalho. Considera o projeto de cargos, as questões de saúde e a interface do usuário nos sistemas de informação.

Os usuários serão mais cooperativos se os problemas organizacionais forem resolvidos antes da introdução do novo sistema. Além das mudanças nos procedimentos, as transformações nas funções de cada cargo, na estrutura organizacional, nas relações de poder e no comportamento devem ser identificadas durante a análise de sistemas por meio de uma **análise de impacto organizacional**.

A seção "Pessoas em destaque" ilustra algumas dessas estratégias de implementação na prática. O governo australiano desenvolveu um novo sistema para o Centrelink, o gigantesco órgão de assistência social do país, para proporcionar a prestação eficiente de serviços sociais on-line. Para desenvolver esse sistema com sucesso, a CIO do Centrelink, Jane Treadwall, teve de superar desafios técnicos e organizacionais. Ao ler este estudo de caso, procure identificar o problema enfrentado pela organização; quais eram as soluções disponíveis à administração; em que medida a solução escolhida foi apropriada; e as questões humanas, organizacionais e tecnológicas que tiveram de ser abordadas durante o desenvolvimento da solução.

**11.3 Diferentes abordagens de desenvolvimento de sistemas**

Existem diferentes métodos para desenvolver sistemas usando o modelo básico de resolução de problemas que acabamos de descrever. Entre esses métodos estão o ciclo de vida de sistemas tradicional, o desenvolvimento pelo usuário final, os pacotes de software aplicativo e o outsourcing (terceirização).

**Ciclo de vida de sistemas tradicional**

O **ciclo de vida de sistemas** é o método mais antigo de montagem de sistemas de informação. A metodologia do ciclo de vida é uma abordagem em fases à montagem de um sistema, dividindo o desenvolvimento de sistemas em estágios formais, conforme ilustrado na Figura 11.5. Embora os desenvolvedores de sistema possam ir e voltar entre um estágio e outro, o ciclo de vida de sistemas é predominantemente uma abordagem 'em cascata', na qual as tarefas de cada estágio devem estar concluídas antes que o estágio seguinte se inicie.

PLANO CONJUGADO SIRH — RH (SIRH — SISTEMA DE INFORMAÇÃO DE RECURSOS HUMANOS)	Dado	Resp.	2006			2007									2008					
			Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez	Jan	Fev	Mar
<b>SEGURANÇA DO GERENCIAMENTO DE DADOS</b>																				
Revisão/montagem de segurança QMF	20	EF TP																		
Orientação de segurança	2	EF JV																		
Manutenção de segurança QMF	35	TP GL																		
Perfis seg. entrada de dados	4	EF TP																		
Estim. visões seg. entrada de dados	12	EF TP																		
Perfis de segurança entrada de dados	65	EF TP																		
<b>DICIONÁRIO DE DADOS</b>																				
Sessões de orientação	1	EF																		
Projeto de dicionário de dados	32	EF WV																		
Coord. cons. prod. DD	20	GL																		
Coord. ao vivo prod. DD	40	EF GL																		
Depuração dicionário de dados	35	EF GL																		
Manut. dicionário de dados	35	EF GL																		
<b>REVISÃO DE PROCEDIMENTOS NA PREPARAÇÃO DO PROJETO</b>																				
Fluxos de trabalho (antigos)	10	PK JL																		
Fluxos de dados folha de pagamento	31	JL PK																		
Modelo SIRH	11	PK JL																		
Interface orientada p/ marketing	6	PK JL																		
Interface coord. 1	15	PK																		
Interface coord. 2	8	PK																		
Interfaces de benefícios (antigas)	5	JL																		
Novo fluxo interfaces de benef.	8	JL																		
Estratégia de comunicação de benef.	3	PK JL																		
Modelo novo do fluxo trabalho	15	PK JL																		
Fluxos de entrada de dados	14	WV JL																		
<b>RESUMO DE RECURSOS</b>																				
Edith Farrell	5.0	EF																		
Woody Vinton	5.0	WV																		
Charles Pierce	5.0	CP	2	21	24	24	23	22	22	27	34	34	29	26	28	19	14			
Ted Leurs	5.0	TL	5	17	20	19	12	10	14	10	2							4	3	
Toni Cox	5.0	TC		5	11	20	13	9	10	7	6	8	4	4	4	4	4			
Patricia Knopp	5.0	PK		12	17	17	19	17	14	12	15	16	2	1	1	1	1			
Jane Lawton	5.0	JL	1	11	10	11	11	12	19	19	21	21	21	17	17	12	9			
David Holloway	5.0	DH	7	23	30	34	27	25	15	24	25	16	11	13	17	10	3	3	2	
Diane O'Neill	5.0	DO	1	9	16	21	19	21	21	20	17	15	14	12	14	8	5			
Joan Albert	5.0	JA	4	4	5	5	5	2	7	5	4	16	2							
Marie Marcus	5.0	MM	6	14	17	16	13	11	9	4										
Don Stevens	5.0	DS	5	6			7	6	2	1				5	5	1				
Casual	5.0	CASL	15	7	2	1	1													
Kathy Mendez	5.0	KM	4	4	5	4	5	1												
Anna Borden	5.0	AB		3	4	3			4	7	9	5	3	2						
Gail Loring	5.0	GL		1	5	16	20	19	22	19	20	18	20	11	2					
NÃO DESIGNADOS	0.0	X					9	10	16	15	11	12	19	10	7	1				
Coop.	5.0	CO =		3	6	5	9	10	17	18	17	10	13	10	10	7	17			
		COOP.											9		236	225	230	14	13	
Casuais	5.0	CAUL =		6	4				2	3	4	4	2	4	16			216	178	
		CAS.							3	3	3									
<b>TOTAL DE DIAS</b>																				
			49	147	176	196	194	174	193	195	190	181	140	125	358	288	284	237	196	12

Figura 11.4

Ferramentas formais de planejamento e controle ajudam a gerenciar projetos de sistemas de informação. O gráfico de Gantt mostra a tarefa, o número de homens-dia e as iniciais de cada profissional responsável, bem como as datas de início e término de cada tarefa. Com o resumo de recursos, um bom gerente consegue ver o total de homens-dia para cada mês e para cada profissional que trabalha no projeto.

Essa abordagem mantém uma divisão de trabalho muito formal entre usuários finais e especialistas em sistemas de informação. Técnicos especializados, como analistas de sistemas e programadores, são responsáveis por grande parte da análise, projeto e trabalho de implementação dos sistemas; os usuários finais limitam-se a apresentar seus

## PESSOAS EM DESTAQUE

### Mudanças na administração dos serviços sociais australianos

Em 1997, o recém-eleito governo conservador da Austrália resolveu remodelar o serviço de assistência social do país. Naquela época, praticamente um terço dos 20 milhões de cidadãos australianos estava recebendo algum tipo de benefício: ou porque estavam desempregados, ou porque eram aposentados, pais solteiros, deficientes, estudantes, membros da população indígena, ou porque se encaixavam em outras categorias elegíveis. Dois órgãos públicos atendiam a grande maioria desses beneficiados: o Departamento de Trabalho, Educação, Treinamento e Assuntos da Juventude (DEETYA) e o Departamento de Serviços Sociais (DSS). A principal função do DEETYA era ajudar a encontrar trabalho para os desempregados por meio de seus aproximadamente 300 escritórios. Já o DSS prestava serviços por meio de outros 300 escritórios regionais. Juntos, os dois departamentos empregavam mais de 30 mil pessoas.

Embora o sistema de benefícios do DSS fosse amplo, era ineficiente e nada amigável ao cidadão. O típico escritório DSS tinha uma equipe de 65 funcionários e atendia centenas de pessoas diariamente, cobrindo uma variedade de tipos de benefício. O processo pelo qual um cidadão protocolava um pedido parecia mais uma tentativa de depurar a base de beneficiários do que de prestar serviços. O DSS submetia os candidatos a uma série de entrevistas, avaliações e revisões para assegurar que seus pedidos estivessem completos e precisos, assim como sua elegibilidade. Os funcionários do departamento não tomavam nenhuma decisão na primeira visita do demandante e, na maioria das vezes, quem ia pela primeira vez tinha de voltar de qualquer modo, pois não trazia todas as informações exigidas.

Quando a administração do primeiro-ministro John Howard tomou posse, uma de suas primeiras iniciativas foi, em um prazo de cinco anos, transformar a experiência inamistosa e muitas vezes irritante de visitar um escritório do DEETYA ou do DSS num e-business único e amigável. Entre os obstáculos estavam uma rivalidade de longa data entre os dois departamentos e uma complicada fusão de seus milhares de funcionários. O novo órgão foi batizado de Centrelink.

Para preencher o cargo de *chief information officer* (CIO), a CEO Sue Vardon não queria um especialista técnico, e sim uma mente excepcional. Ela encontrou o que buscava em Jane Treadwell, sua ex-colega no Departamento de Serviços Correccionais do Sul da Austrália. Vardon valorizava a inteligência de Treadwell e sua habilidade para prever o impacto futuro das decisões do presente. Ela trouxe uma perspectiva empresarial para um órgão público que tinha um problema de TI. Preparando-se para o desafio, Treadwell estudou gerenciamento de mudanças e visitou órgãos públicos ao redor do mundo. Lançou-se, então, à tarefa de convencer os tradicionalistas do governo de que os sistemas de informação não eram um sumidouro de dinheiro, e que os 312 milhões de dólares pagos pelo Centrelink trariam dividendos.

Para atingir suas metas, que incluíam canais de auto-atendimento por telefone e Internet para transações de benefícios, o Centrelink não podia depender de soluções

pré-fabricadas. A organização desenvolveu seu próprio *middleware* customizado, denominado Centrelink Online Framework (COLF). Para criar produtos de auto-atendimento viáveis, a equipe de sistemas de informação teve de traduzir mais de 500 códigos de programa de software, extraídos de tabelas de dicionários de dados nos sistemas legados da organização, em menus ou telas de diálogo apropriados a softwares de reconhecimento de voz e em formulários amigáveis para páginas Web. Os canais de telefone e Internet foram projetados para imitar a interação entre um atendente de carne e osso e um cliente do Centrelink. Os consumidores podem revisar e atualizar suas contas facilmente, com a conveniência extra do anonimato. O Centrelink processa as informações e resolve pedidos e reclamações mais rapidamente do que seus antecessores.

A satisfação do cliente com o Centrelink subiu 23 pontos percentuais desde a chegada de Treadwell, chegando a 86 por cento. Em 2002, o Boston Consulting Group verificou um aumento de 21 por cento na produtividade ao longo dos primeiros cinco anos de existência do Centrelink. Boa parte do mérito vai para as técnicas organizacionais e a administração de Treadwell. Ela mudou a percepção de TI ao incorporá-la à alta administração e à esfera dos negócios. Sob sua liderança, não havia mais projetos de TI, apenas projetos de negócios. Em consequência, Treadwell acrescentou ao título de CIO o de CEO Adjunta de Transformação Empresarial.

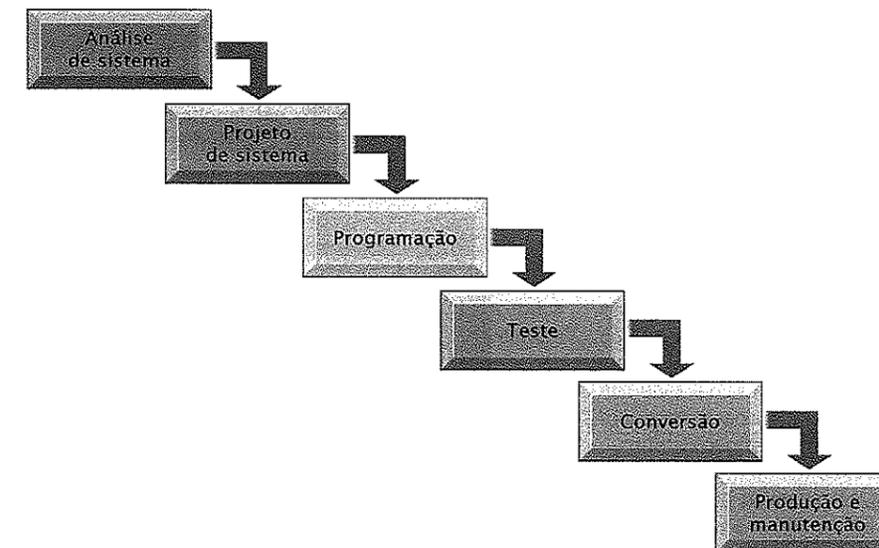
Treadwell também foi inovadora em outros aspectos. Ciente da importância de administrar a mudança na empresa cuidadosamente, ela criou a Coalizão Orientadora, formada por todos os 60 executivos do Centrelink. A Coalizão se reunia a cada seis ou oito semanas para verificar como a tecnologia da informação poderia promover novas oportunidades em serviços. Treadwell também eliminou os cargos tradicionais em seu grupo de transformação empresarial. Lá, os funcionários são considerados parte de um 'banco de talentos'. Designados para papéis e projetos específicos, são realocados para outros quando os projetos terminam. O objetivo é evitar a estagnação que os cargos permanentes podem causar.

Hoje, o Centrelink opera a segunda maior rede de *call center* da Austrália. Seus mil escritórios supervisionam benefícios que totalizam 55 bilhões de dólares australianos, provenientes de 25 órgãos públicos. Tendo atingido suas metas, Treadwell deixou o Centrelink na primavera de 2005 para se tornar CIO do governo de Victoria.

Fontes: Lauraine Sayers, "Aussie whirlwind sweeps through government IT", *CIO Insight*, [www.cioinsight.com](http://www.cioinsight.com), acessado em 05 maio 2005.

#### Para pensar:

Quais problemas os sistemas de assistência social enfrentavam na Austrália? Como o Centrelink tentou solucionar esses problemas? Que outras soluções poderiam ter funcionado? Que fatores humanos, organizacionais e tecnológicos afetavam esse problema e sua solução? Descreva o papel de Jane Treadwell no desenvolvimento do Centrelink. Quais escolhas e técnicas contribuíram para o seu sucesso como executiva?



**Figura 11.5**

Ciclo de vida de sistemas tradicional. O ciclo de vida de sistemas fragmenta o desenvolvimento de sistemas em estágios formais; cada estágio precisa ser finalizado antes que o seguinte se inicie.

requisitos de informação e revisar o trabalho do pessoal técnico. O ciclo de vida dá ênfase a especificações formais e a papéis; portanto, muitos documentos são gerados no decurso de um projeto de sistemas.

O ciclo de vida de sistemas também é usado no desenvolvimento de grandes sistemas complexos, que exigem uma análise rigorosa e formal de requisitos, especificações predefinidas e controles rígidos sobre o processo de montagem de sistemas. Todavia, a abordagem de ciclo de vida de sistemas é dispendiosa e demorada. As tarefas de um estágio devem estar concluídas antes que o trabalho do estágio seguinte se inicie. As atividades podem ser refeitas, mas grandes volumes de novos documentos devem ser gerados e etapas precisam ser repetidas, caso seja necessário revisar requisitos e especificações. Isso estimula o congelamento das especificações em uma etapa relativamente precoce do processo de desenvolvimento. A abordagem de ciclo de vida também é inadequada para pequenos sistemas de computadores de mesa, que tendem a ser menos estruturados e mais individualizados.

### Prototipagem

A **prototipagem** consiste em montar um sistema experimental rapidamente e sem muitos gastos para submetê-lo à avaliação de usuários finais. O protótipo é uma versão funcional de um sistema de informação, ou de parte dele, mas deve ser considerado apenas um modelo preliminar. Os usuários interagem com o protótipo para ter uma idéia melhor de seus requisitos de informação, podendo refiná-lo inúmeras vezes. O caso de abertura deste capítulo descreve como a Dovetail Associates usou o QuickBase para criar um protótipo, com o qual o Patriots' Trail Girl Scout Council pôde refinar as especificações para seu sistema de pedidos de biscoito. Assim que o projeto for concluído, o protótipo será convertido em um sistema em produção, refinado e acabado. A Figura 11.6 mostra um modelo de quatro etapas para o processo de prototipagem.

**Etapa 1: Identificação dos requisitos básicos do usuário.** O projetista de sistemas (usualmente um especialista em sistemas de informação) trabalha com o usuário apenas o tempo suficiente para captar suas necessidades básicas de informação.

**Etapa 2: Desenvolvimento de um protótipo inicial.** O projetista de sistemas elabora um protótipo funcional rapidamente, usando ferramentas para gerar softwares com rapidez.

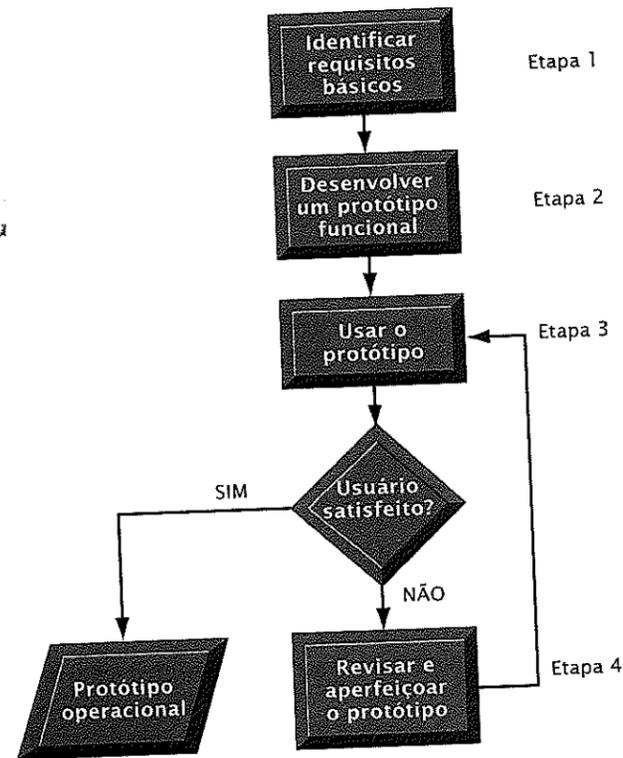
**Etapa 3: Utilização do protótipo.** O usuário é incentivado a trabalhar com o sistema, a fim de determinar em que medida o protótipo atende às suas necessidades e de fazer sugestões para aperfeiçoá-lo.

**Etapa 4: Revisão e aperfeiçoamento do protótipo.** O desenvolvedor do sistema anota todas as alterações requisitadas pelo usuário e refina o protótipo de acordo com essas requisições. Após a revisão do protótipo, o ciclo retorna à etapa 3. As etapas 3 e 4 são repetidas até que o usuário se dê por satisfeito.

A prototipagem é especialmente útil para o projeto da interface de usuário do sistema de informação. Como ela favorece intensa participação do usuário final durante todo o processo de desenvolvimento do sistema, aumenta a probabilidade de que seus requisitos sejam satisfeitos.

**Figura 11.6**

Processo de prototipagem. O processo de desenvolvimento de um protótipo pode ser desdobrado em quatro etapas. Uma vez que é possível desenvolver um protótipo rapidamente e sem grandes custos, os desenvolvedores de sistemas podem passar por diversas iterações, repetindo as etapas 3 e 4 para refiná-lo e aperfeiçoá-lo antes de chegar ao protótipo operacional final.



Todavia, a prototipagem rápida pode negligenciar etapas essenciais no desenvolvimento de sistemas, tais como teste e documentação. Se o protótipo concluído funcionar razoavelmente bem, a administração poderá não perceber a necessidade de montar um sistema de produção refinado. Alguns desses sistemas construídos às pressas podem não acomodar facilmente grandes quantidades de dados ou grande número de usuários em um ambiente de produção.

A seção "Organizações em destaque", a seguir, descreve como a Marriott Corporation usou a prototipagem ao desenvolver seu sistema de gerenciamento de receita One Yield. A empresa precisava de um novo sistema que determinasse a diária ideal a cobrar por cada quarto em todos os seus hotéis. Ao ler este estudo de caso, procure determinar o problema enfrentado pela empresa; as soluções disponíveis à administração; o papel da prototipagem no desenvolvimento da solução; em que medida a solução escolhida foi apropriada; e as questões humanas, organizacionais e tecnológicas que tiveram de ser abordadas durante o desenvolvimento da solução.

### Desenvolvimento pelo usuário final

No desenvolvimento pelo usuário final, os usuários finais, com pouca ou nenhuma assistência formal de técnicos especializados, podem criar sistemas de informação simples, reduzindo o tempo e as etapas necessárias para produzir uma aplicação acabada (veja a Figura 11.7). Utilizando linguagens de quarta geração, linguagens gráficas e ferramentas de software para PCs, os usuários finais podem acessar dados, criar relatórios e desenvolver sistemas de informação inteiros por si mesmos, com pouca ou nenhuma assistência de analistas de sistemas ou programadores profissionais.

Por exemplo, a Administaff, que presta serviços de recursos humanos para cinco mil empresas de pequeno e médio porte em todos os Estados Unidos, usa o software WebFocus, da InformationBuilders Inc., para dar a seus clientes acesso aos dados de seus funcionários. Os clientes usam o WebFocus Studio Developer e outras ferramentas para desenvolver seus próprios relatórios e visualizar seus dados a partir de um repositório central da Administaff. O gerente de uma das empresas clientes da Administaff, por exemplo, pode usar essas ferramentas para criar um relatório mostrando quantos funcionários participam do plano de previdência da empresa (Babcock, 2005).

De modo geral, os sistemas desenvolvidos por usuários finais são finalizados muito mais rapidamente do que os desenvolvidos com as ferramentas convencionais de programação. Permitir que os usuários especifiquem suas próprias necessidades empresariais melhora a coleta de requisitos e freqüentemente leva a um nível mais alto de envolvimento e satisfação do usuário com o sistema. Contudo, ferramentas de quarta geração ainda não podem

## ORGANIZAÇÕES EM DESTAQUE O novo sistema de gerenciamento de receita da Marriott é um sucesso

A Marriott International, Inc. é uma empresa de 9 bilhões de dólares que administra 2.600 hotéis no mundo todo. Como outras cadeias de hotéis, para maximizar a receita a Marriott precisa manter seus quartos ocupados com hóspedes que paguem as mais altas diárias possíveis.

Determinar quanto deveria ser cobrado por quarto costumava ser um grande desafio. Em noites de alta demanda, a Marriott precisava tomar cuidado para não ocupar todos os quartos cedo demais a diárias muito baixas. Nos domingos e nos dias mais parados, precisava ter promoções suficientes para atrair clientes, mas não tantas a ponto de ocupar os quartos que poderiam ser oferecidos a uma taxa mais alta a outras pessoas.

A Marriott havia desenvolvido dois sistemas de gerenciamento de receita. Um sistema de previsão de demanda atendia os hotéis de serviços completos da cadeia, e um sistema de gerenciamento da receita era usado pelos hotéis de permanência longa ou de serviço selecionado, como os Hotéis Courtyard. No entanto, ter dois sistemas independentes de gerenciamento de receita saía caro, sem falar que a cadeia de hotéis tinha dificuldades para analisar as 62 milhões de reservas anuais feitas em todos os seus estabelecimentos.

Em 2001, a Marriott começou a desenvolver um novo sistema de gerenciamento de receita, válido para toda a companhia: o One Yield, hoje usado por 1.700 dos 2.600 estabelecimentos da rede. Desde o início, os líderes do projeto vincularam o valor empresarial do One Yield às metas corporativas de lucratividade, preferência e crescimento. Eles explicaram os aspectos técnicos do sistema em termos que os empresários pudessem entender e avaliar.

Para obter os requisitos de informação e apoio ao novo sistema, a equipe de desenvolvimento consultou os futuros usuários e os compradores do One Yield (cada estabelecimento da Marriott paga por suas próprias instalações de TI). A equipe manteve estreita comunicação de campo por meio do diretor de planejamento de estoque da Marriott, Russell Verb, e conseguiu gerar uma valiosa troca de idéias com usuários e gerentes gerais. Esses gerentes podiam fazer perguntas sobre custos e impacto na equipe, enquanto os usuários consultavam sobre as novas funções do sistema, ou sobre funções antigas que deixariam de existir. Mensalmente, Verb distribuía press releases com as novidades quanto ao andamento do projeto, testes piloto, ferramentas de treinamento e custos.

Por indicação dos usuários, foram incorporadas duas importantes funções ao One Yield: um botão de atalho e aquilo que os usuários chamaram de 'chefão das manhãs de segunda'. O botão de atalho permite que o usuário rejeite as recomendações de diárias do One Yield antes que elas sejam enviadas ao sistema central de reservas. Nos casos em que o gerente de receita local possui informações que o One Yield não tem, tais como uma súbita mudança climática que pode causar cancelamentos, esse recurso é essencial para maximizar os lucros. O 'chefão das manhãs

de segunda' é um monitor de desempenho que consegue analisar quatro semanas de decisões de reserva, a fim de verificar em que medida o gerente de receita se aproximou da receita ideal durante o período selecionado. Essa função também informa quais decisões o gerente deveria ter tomado para maximizar os lucros.

Outra chave para o sucesso do One Yield foi a prototipagem precoce. O departamento de sistemas de informação da Marriott começou desenvolvendo protótipos dos módulos mais importantes do sistema, enquanto a equipe administrativa ainda estava montando o estudo de caso. A prototipagem precoce garantiu que o lançamento do sistema — que só ocorreria três anos depois — transcorresse sem tropeços. Ao trabalhar com o protótipo, a equipe de desenvolvimento de sistemas resolveu problemas como relatórios que levavam 5 minutos para serem executados, quando deveriam levar 5 segundos.

A equipe de projetos havia decidido desenvolver internamente o sistema como uma aplicação baseada na Web, usando arquitetura J2EE e ferramentas de desenvolvimento WebSphere. Ninguém na Marriott havia trabalhado com essa tecnologia antes. As únicas experiências da empresa com a Web tinham sido projetos de pequena escala, com exibições estáticas de conteúdo, como páginas Web. A prototipagem deu à equipe do One Yield familiaridade com a tecnologia. Embora essa metodologia tenha atrasado o lançamento inicial do One Yield em dois meses, ela provavelmente economizou à Marriott muito tempo e dinheiro que teriam sido gastos para resolver problemas — os quais certamente surgiriam caso não tivesse existido essa fase experimental.

O One Yield recuperou 80 por cento de seus custos pouco tempo depois de lançado. Em 2004, os estabelecimentos da Marriott que usavam o sistema experimental um aumento de 2 por cento na receita proveniente de turistas, o que contribuiu para um aumento no lucro anual de 86 milhões de dólares. A empresa como um todo vem obtendo um aumento significativo na receita operacional desde a implantação do One Yield.

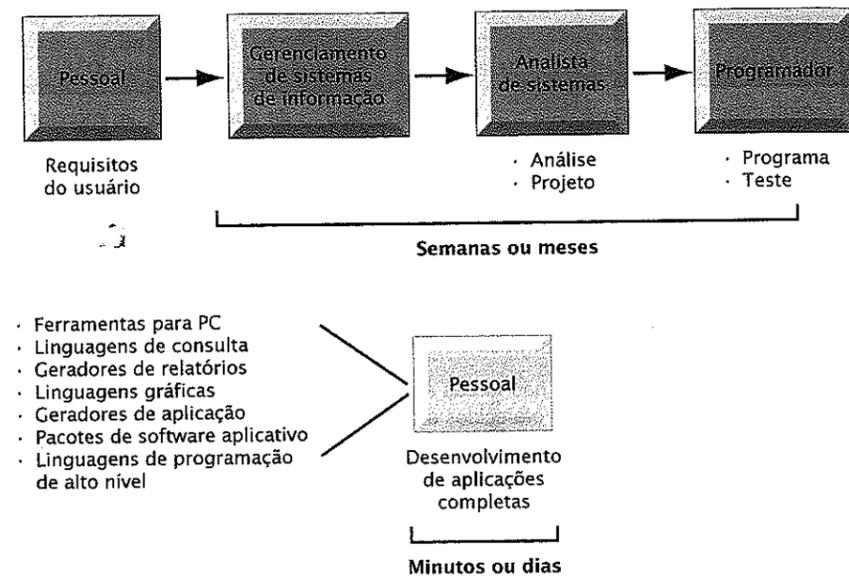
Fontes: Stephanie Overby, "The price is always right", *CIO Magazine*, www.cio.com, acessado em 15 fev. 2005; "Marriott named a 'best place to work in information technology' by *Computerworld* for third consecutive year", *Hotel News Resource*, www.hotelnewsresource.com, acessado em 29 jun. 2005; e Patrick Thibodeau, "Marriott links two data streams with revenue management system", *Computerworld*, www.computerworld.com, acessado em 14 mar. 2005.

### Para pensar:

Qual o problema enfrentado pela Marriott? Quais objetivos empresariais a companhia estava tentando alcançar? Quais eram as soluções disponíveis para resolver o problema? A empresa escolheu a melhor solução? Como a prototipagem ajudou a Marriott a elaborar sua solução? Quais fatores humanos, organizacionais e tecnológicos estavam envolvidos na solução que a empresa escolheu?

**Figura 11.7**

Desenvolvimento pelo usuário final. Ferramentas de software de quarta geração possibilitam aos usuários finais desenvolver sistemas simples que podem ser concluídos mais rapidamente do que aqueles desenvolvidos por meio do tradicional ciclo de vida de desenvolvimento de sistemas.



substituir ferramentas convencionais, no caso de algumas aplicações empresariais, porque não conseguem administrar facilmente o processamento de grandes números de transações ou aplicações com requisitos completos de lógica e atualização.

A computação por usuário final também apresenta riscos organizacionais porque os sistemas são criados rapidamente, sem metodologia formal de desenvolvimento, teste e documentação. Para ajudar as organizações a maximizar os benefícios do desenvolvimento de aplicações pelo usuário final, a administração deve exigir uma justificação de custos dos projetos de sistemas de informação a serem desenvolvidos pelo usuário final e estabelecer padrões de qualidade, de software e de hardware para esse tipo de aplicação.

### Soluções de compra: pacotes de software aplicativo e outsourcing

No Capítulo 4 ressaltamos que hoje em dia a maior parte dos sistemas usa software que não foi desenvolvido internamente, mas adquirido de fontes externas. As empresas podem escolher entre comprar um pacote de software de um fornecedor comercial, alugar o software de um provedor de serviços de aplicação, ou terceirizar o trabalho de desenvolvimento para outra empresa. A seleção do software ou serviço de software muitas vezes se baseia em uma **requisição formal de proposta (request for proposal)**, que é uma lista detalhada de perguntas apresentadas aos fabricantes do pacote de software para verificar em que medida eles atendem aos requisitos do sistema proposto.

#### Pacotes de softwares aplicativos

Hoje, a maioria dos novos sistemas de informação é desenvolvida com um pacote de software aplicativo ou com componentes de software pré-programados. Muitas aplicações são comuns a todas as organizações empresariais — por exemplo, folha de pagamento, contas a receber, livro-razão ou controle de estoque. Para tais funções universais, cujos procedimentos são padronizados e não mudam muito ao longo do tempo, um sistema generalizado atenderá aos requisitos de muitas organizações.

Se determinado pacote de software conseguir atender à maioria dos requisitos de uma organização, ela não precisará escrever seus próprios programas. Poderá poupar tempo e dinheiro utilizando os programas de software pré-escritos, pré-elaborados e pré-testados do pacote.

Muitos pacotes incluem recursos de customização para atender a requisitos exclusivos não abordados pelo pacote. Características de **customização** permitem que um pacote de software seja modificado para atender aos requisitos exclusivos de uma organização sem destruir sua integridade. Contudo, caso seja necessária uma customização extensa, o trabalho adicional de programação e customização pode ficar tão caro e ser tão demorado que acabará eliminando muitas das vantagens dos pacotes de software. Se o pacote não puder ser customizado, a organização terá de se adaptar ao pacote e alterar seus procedimentos.

### Terceirização (outsourcing)

Se uma empresa não quiser usar seus recursos internos para montar ou operar sistemas de informação, pode contratar uma organização externa especializada em prover esses serviços. Os provedores de serviços aplicativos (ASPs), que descrevemos no Capítulo 4, são uma das formas de terceirização. Exemplo disso seriam as Girl Scouts alugando o hardware e o software do QuickBase para rodar seu sistema de pedidos de biscoito. As empresas assinantes usam o software e o hardware fornecidos pelo ASP como plataforma técnica para seus sistemas. Em outro tipo de terceirização, uma empresa contrata um fabricante externo para projetar e criar o software para seu sistema, mas opera o sistema em seu próprio computador.

O fornecedor de outsourcing pode ser doméstico ou estrangeiro. O outsourcing doméstico é motivado principalmente pelo fato de que as empresas terceirizadas possuem as habilidades, recursos e ativos que seus clientes não têm. Um novo sistema de gerenciamento da cadeia de suprimentos em uma empresa muito grande pode exigir a contratação de 30 a 50 pessoas com conhecimentos especializados em software licenciado de gerenciamento da cadeia de suprimentos fabricado, digamos, pela Manugistics ou outro fornecedor. Em vez de contratar novos funcionários em caráter definitivo, a maior parte deles precisando de extenso treinamento no pacote de software, e depois demiti-los quando o novo sistema estiver concluído, faz mais sentido — e geralmente sai mais barato — terceirizar esse trabalho por um período de 12 meses.

No caso do outsourcing internacional, a decisão tende a ser muito mais motivada pelo custo. Um programador gabaritado na Índia ou na Rússia ganha cerca de 10 mil dólares por ano, contra 70 mil dólares por ano para um programador equivalente nos Estados Unidos. A Internet e o baixo custo das tecnologias de comunicação vêm reduzindo drasticamente o custo e a dificuldade de coordenar o trabalho de equipes globais em lugares distantes. Além de custos mais baixos, muitas empresas de outsourcing internacional oferecem *know-how* e ativos de tecnologia de classe mundial.

Por exemplo, a gigantesca empresa de viagens européia TUI AG optou pelo outsourcing para reduzir seus custos no competitivo mercado de serviços de reserva de viagens. A TUI conseguiu economizar 20 por cento ao contratar o provedor de serviços e software indiano Wipro Ltd. para prestar suporte remoto a cerca de 10 mil sistemas de computador de mesa e 300 servidores departamentais, incluindo help desks e mensagens por e-mail. A TUI agora está terceirizando o desenvolvimento de sistemas customizados e suporte para novas aplicações com a Sonata Software Ltd., outro fornecedor de serviços e software indiano (Ribeiro, 2005).

Existe grande probabilidade de que, em algum ponto da sua carreira, você trabalhe com terceiros internacionais ou equipes globais. Sua empresa terá mais chances de se beneficiar do outsourcing se avaliar com cuidado todos os riscos e tiver certeza de que essa estratégia é adequada a suas necessidades particulares. Qualquer empresa que quiser terceirizar suas aplicações precisa compreender profundamente o projeto, inclusive seus requisitos, método de implementação, fonte de benefícios esperados, componentes de custo e métricas para mensurar o desempenho.

Muitas empresas subestimam os custos de identificar e avaliar prestadores de serviços de tecnologia de informação, de fazer a transição para um novo prestador, de aperfeiçoar os métodos de desenvolvimento de software internos para que se compatibilizem com os dos fornecedores terceirizados e de monitorá-los para garantir o cumprimento de suas obrigações contratuais. O outsourcing internacional inclui ainda custos adicionais para lidar com as diferenças culturais que diminuem a produtividade e com questões de recursos humanos, tais como demitir ou realocar funcionários domésticos. Esses 'custos ocultos' podem facilmente restringir os benefícios que eram esperados da terceirização. As empresas devem ser especialmente cautelosas ao usar um terceiro para desenvolver ou operar aplicações que lhes dêem algum tipo de vantagem competitiva.

A Figura 11.8 mostra os cenários otimista e pessimista para o custo total de um projeto de outsourcing internacional, revelando em que medida os custos ocultos afetam esse custo total. O cenário otimista reflete as estimativas mais baixas para os custos adicionais, enquanto o pessimista reflete as mais altas estimativas para esses custos. Como você pode ver, os custos ocultos aumentam o custo total de um projeto de outsourcing internacional de 15 a 57 por cento. Mesmo com esses custos extras, muitas empresas podem beneficiar-se do outsourcing internacional caso gerenciem bem o trabalho.

### Desenvolvimento rápido de aplicações para e-business

As tecnologias e condições de negócios estão mudando tão rapidamente que agilidade e escalabilidade se tornaram elementos críticos de soluções de sistemas. As empresas vêm adotando processos de desenvolvimento mais curtos e informais para muitas de suas aplicações de e-commerce e e-business, processos que oferecem soluções rápidas sem causar rupturas em seus sistemas transacionais centrais e bancos de dados organizacionais. Além de

Custo total do outsourcing internacional				
\$10.000.000				
Custo do contrato de outsourcing				
Custos ocultos	Cenário otimista	Custo adicional (\$)	Cenário pessimista	Custo adicional (\$)
1. Seleção de fornecedores	0,2%	20.000	2%	200.000
2. Custos de transição	2%	200.000	3%	300.000
3. Demissões e retenção	3%	300.000	5%	500.000
4. Perda de produtividade/problemas culturais	3%	300.000	27%	2.700.000
5. Aperfeiçoamento de processos de desenvolvimento	1%	100.000	10%	1.000.000
6. Gestão do contrato	6%	600.000	10%	1.000.000
<b>Custos adicionais totais</b>		<b>1.520.000</b>		<b>5.700.000</b>
	Contrato em aberto (\$)	Custo adicional (\$)	Custo total (\$)	Custo adicional
Custo total do outsourcing no cenário otimista	10.000.000	1.520.000	11.520.000	15,2%
Custo total do outsourcing no cenário pessimista	10.000.000	5.700.000	15.700.000	57,0%

Figura 11.8

Custo total do outsourcing internacional. Se uma empresa gastar 10 milhões de dólares em contratos de outsourcing internacional, essa empresa na verdade gastará 15,2 por cento a mais em custos extras, mesmo no cenário mais otimista. No cenário mais pessimista, em que existe uma queda drástica na produtividade, combinada com custos de demissão e transição excepcionalmente altos, a empresa pode esperar desembolsar 57 por cento a mais em custos extras, além dos 10 milhões pagos pelo contrato internacional.

usar pacotes de software, provedores de serviços aplicativos e outros serviços de outsourcing, elas estão confiando mais em técnicas de ciclo rápido, como o projeto conjunto de aplicações (*joint application design* — JAD), protótipos e componentes de software padronizados e reutilizáveis que podem ser montados como um conjunto completo de serviços para e-commerce e e-business.

O termo **desenvolvimento rápido de aplicações** (*rapid application development* — RAD) refere-se ao processo de criar sistemas aptos a funcionar em um espaço muito curto de tempo. O RAD inclui o uso de programação visual e outras ferramentas para desenvolver interfaces gráficas de usuário, prototipagem iterativa de elementos de sistema essenciais, automação da geração de códigos-fonte para programas e colaboração mais estreita entre usuários finais e especialistas em sistemas de informação. Sistemas simples muitas vezes podem ser montados a partir de componentes pré-construídos (veja a Seção 11.4). O processo não precisa ser seqüencial, e partes-chave do desenvolvimento podem ocorrer simultaneamente.

Às vezes, uma técnica denominada **projeto conjunto de aplicações** (*joint application design* — JAD) é usada para acelerar a geração de requisitos de informação e desenvolver o projeto inicial de sistemas. A JAD reúne usuários finais e especialistas em sistemas de informação em uma sessão interativa para discutir o projeto do sistema. Se adequadamente preparadas e mediadas, as sessões JAD podem acelerar significativamente a fase de projeto e envolver intensamente os usuários.

### 11.4 Como formatar e projetar sistemas: metodologias estruturadas e orientadas a objetos

Acabamos de descrever diferentes métodos para o desenvolvimento de sistemas. Também existem diferentes alternativas para a formatação e o projeto de sistemas. As duas mais importantes são as metodologias estruturadas e o desenvolvimento orientado a objetos.

#### Metodologias estruturadas

Metodologias estruturadas vêm sendo utilizadas para documentar, analisar e projetar sistemas de informação desde a década de 1970. São chamadas **estruturadas** as técnicas elaboradas passo a passo, sendo cada um dos passos fundamentado no anterior. As metodologias estruturadas são executadas de cima para baixo, partindo-se do nível mais alto e mais abstrato para o nível mais baixo de detalhe — do geral para o específico.

Os métodos de desenvolvimento estruturados são orientados a processos, concentrando-se primordialmente em formatar os processos, ou ações, que capturam, armazenam, manipulam e distribuem dados, como o fluxo de dados ao longo do sistema. Esses métodos distinguem dados de processos. Um procedimento de programação independente precisa ser escrito cada vez que alguém quer fazer algo com determinado conjunto de dados. Os procedimentos atuam sobre os dados que o programa repassa a eles.

A ferramenta primária para representar os processos componentes de um sistema e as interfaces (fluxo de dados) entre eles é o **diagrama de fluxo de dados** (*data flow diagram* — DFD). O diagrama de fluxo de dados oferece um modelo gráfico lógico do fluxo de informações, dividindo um sistema em módulos que mostram os níveis de detalhe administráveis. Ele especifica rigorosamente os processos ou transformações que ocorrem dentro de cada módulo e as interfaces existentes entre eles.

A Figura 11.9 mostra um diagrama de fluxo de dados simples para um sistema de matrícula em um curso universitário a distância. Os retângulos de cantos arredondados representam processos que retratam a transformação de dados. O quadrado representa uma entidade externa que origina ou recebe informações localizadas fora das fronteiras do sistema que está sendo modelado. Os retângulos pontilhados e abertos representam armazéns de dados, isto é, estoques de dados manuais ou automatizados. As setas representam os fluxos de dados que mostram a movimentação entre os processos, as entidades externas e os armazéns de dados. Elas sempre contêm pacotes de dados com o nome ou conteúdo de cada fluxo listado ao lado da seta.

Esse diagrama de fluxo de dados mostra que os estudantes apresentam formulários de matrícula com seus nomes, números de identificação e os números dos cursos que querem frequentar. No processo 1.0, o sistema verifica se cada curso selecionado ainda está aberto, consultando o arquivo de cursos da universidade. O arquivo diferencia cursos que estão abertos daqueles que foram cancelados ou já estão lotados. O processo 1.0 então determina quais das opções do estudante podem ser aceitas ou rejeitadas. Em seguida, o processo 2.0 matricula o estudante nos cursos para os quais ele foi aceito. Também atualiza o arquivo de cursos da universidade com o nome e o número de identificação do estudante e recalcula a lotação da classe. Quando o número máximo de matrículas é alcançado, o número do curso recebe uma marcação (um 'flag') indicando que ele está fechado. O processo 2.0 atualiza ainda o arquivo mestre de alunos da universidade com informações de novos estudantes ou mudanças de endereço. O processo 3.0 então envia a cada estudante inscrito uma carta de confirmação de matrícula, listando os cursos para os quais ele está matriculado e citando as opções que não puderam ser atendidas.

Por meio dos diagramas de fluxo de dados, um processo complexo pode ser desdobrado em sucessivos níveis de detalhe. Com um diagrama de fluxo de alto nível, um sistema inteiro pode ser dividido em subsistemas. Cada subsistema, por sua vez, pode ser dividido em subsistemas adicionais com um diagrama de fluxo de segundo nível, e os subsistemas de nível mais baixo podem ser desdobrados novamente até que seja atingido o nível mais baixo de detalhe. As **especificações de processo** descrevem a transformação que ocorre no nível mais baixo do fluxograma de dados. Expressam a lógica de cada processo.

Na metodologia estruturada, o projeto de software é formatado por meio de um diagrama de estrutura hierárquica. O **diagrama de estrutura** é um diagrama de cima para baixo que mostra cada nível do projeto, sua relação com os outros níveis e sua localização na estrutura geral do projeto. O projeto considera primeiro a função principal

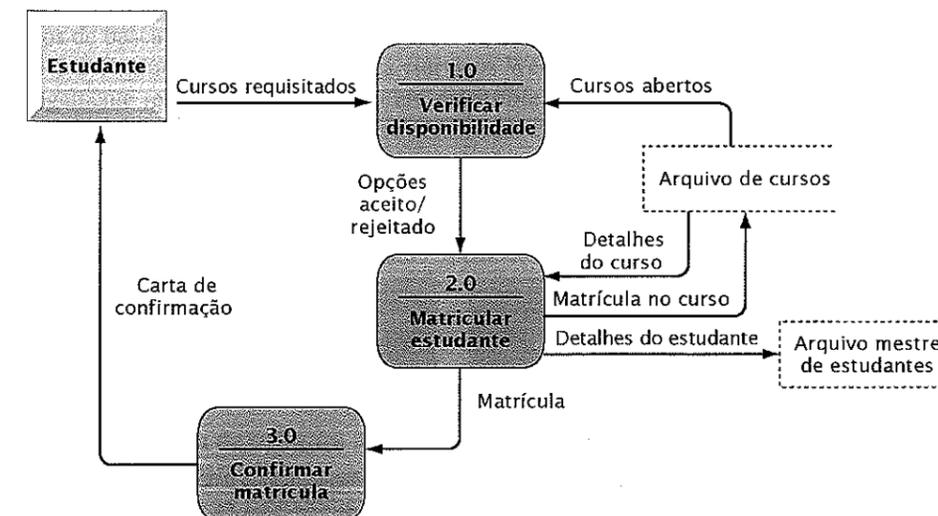


Figura 11.9

Diagrama de fluxo de dados para o sistema de matrícula em uma universidade a distância. O sistema tem três processos: verificar disponibilidade (1.0), matricular estudante (2.0) e confirmar matrícula (3.0). O nome e conteúdo de cada fluxo de dados aparecem adjacentes a cada seta. Há uma entidade externa neste sistema: o estudante. E dois armazéns de dados: o arquivo mestre de estudantes e o arquivo do curso.

do programa ou sistema, então divide esta função em subfunções e as decompõe em níveis de detalhamento de mais baixo nível. A Figura 11.10 mostra um diagrama de estrutura de alto nível para um sistema de folha de pagamento. Se o número de níveis de um projeto for muito grande para ser representado em um único diagrama de estrutura, ele pode ser desdobrado em mais diagramas detalhados. Um diagrama de estrutura pode documentar um programa, um sistema (um conjunto de programas) ou parte de um programa.

### Desenvolvimento orientado a objetos

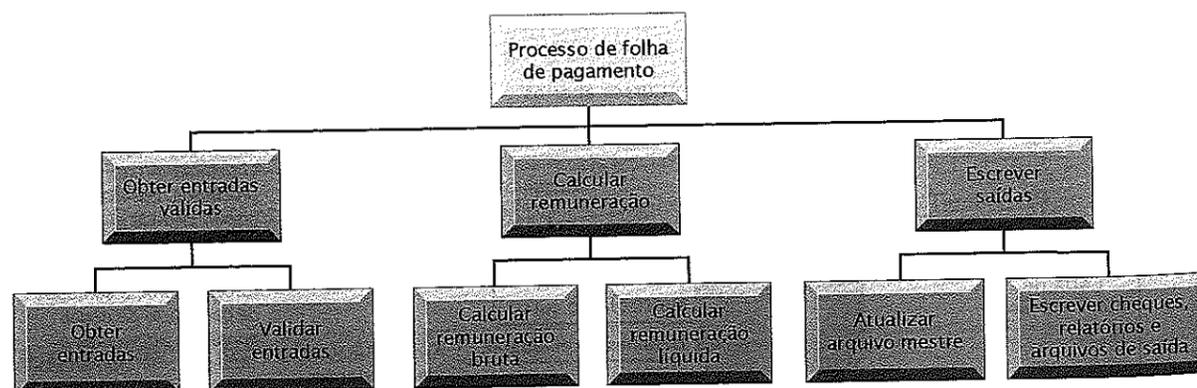
Os métodos estruturados tratam dados e processos como entidades logicamente separadas, mas no mundo real tal separação parece pouco natural. Diferentes convenções de formatação são usadas para análise (o diagrama de fluxo de dados) e projeto (o diagrama de estrutura).

O **desenvolvimento orientado a objetos** aborda essas questões. Tal desenvolvimento usa o objeto, conforme conceituado no Capítulo 4, como a unidade básica da análise e projeto de sistemas. Um objeto combina dados e os processos específicos que operam nesses dados. Os dados encapsulados num objeto podem ser acessados e modificados somente pelas operações — ou métodos — associadas àquele objeto. Em vez de passar os dados para os procedimentos, os programas enviam uma mensagem para que o objeto execute o procedimento nele embutido. O sistema é modelado como um conjunto de objetos e segundo os relacionamentos entre eles. Como a lógica de processamento reside dentro dos objetos, e não em programas de software independentes, os objetos precisam colaborar entre si para fazer o sistema funcionar.

A modelagem orientada a objetos baseia-se em conceitos de *classe* e *herança*. Objetos pertencentes a certa classe — isto é, categorias gerais de objetos similares — têm as características daquela classe. As classes de objetos, por sua vez, podem herdar toda a estrutura e os comportamentos de uma classe mais geral e então adicionar variáveis e comportamentos exclusivos a cada objeto. Novas classes de objetos são criadas escolhendo-se uma classe já existente e especificando quanto a nova classe difere dessa, em vez de começar do zero a cada ocorrência.

Podemos ver o funcionamento das classes e da herança na Figura 11.11, que ilustra as relações entre classes referentes aos funcionários de uma empresa e ao modo como eles são remunerados. Funcionário é o ancestral comum (ou superclasse) para as outras três classes. Mensalista, Horista e Temporário são subclasses de Funcionário. O nome da classe está na parte superior do bloco, os atributos de cada classe ficam na parte do meio, e a lista de operações se localiza na parte inferior do bloco. As características compartilhadas por todos os funcionários (documento de identidade, nome, endereço, data de admissão, cargo e remuneração) são armazenadas na superclasse Funcionário, enquanto cada subclasse armazena características específicas daquele tipo particular de funcionário. A remuneração horária e a remuneração por horas extras, por exemplo, são específicas dos funcionários horistas. As setas que vão das subclasses para a superclasse são um caminho de generalização; elas nos mostram que as subclasses Mensalista, Horista e Temporário têm características comuns que podem ser generalizadas na superclasse Funcionário.

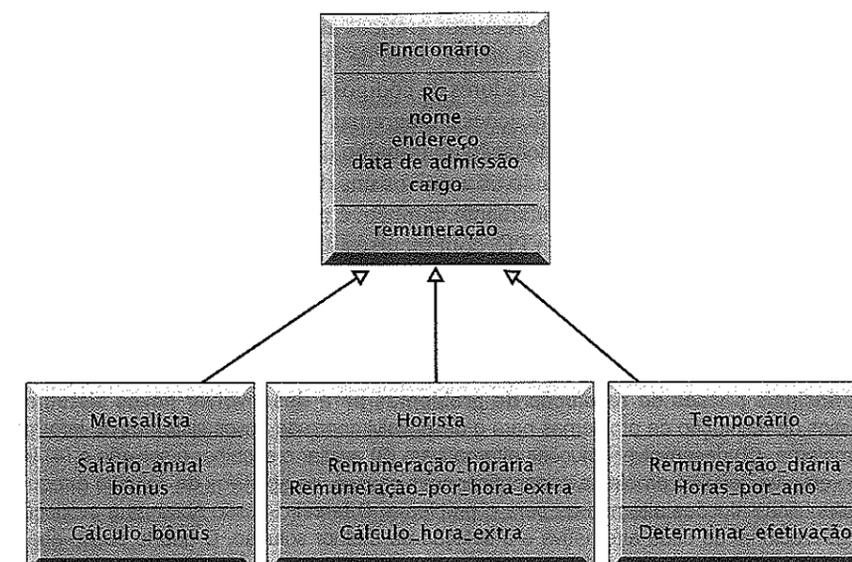
O desenvolvimento orientado a objetos é mais iterativo e incremental do que o desenvolvimento estruturado tradicional. Durante a análise de sistemas, os desenvolvedores de sistemas documentam os requisitos funcionais do sistema, especificando suas propriedades mais importantes e o que o sistema proposto deve fazer. Esses desenvolve-



**Figura 11.10**

Diagrama de estrutura de alto nível para um sistema de folha de pagamento.

Este diagrama mostra o nível mais abstrato, ou mais alto, do projeto de um sistema de folha de pagamento, dando uma visão geral de todo o sistema.



**Figura 11.11**

Classe e herança. Esta figura ilustra como as classes herdam as características comuns de sua superclasse.

dores analisam as interações entre o sistema e seus usuários para identificar objetos, o que inclui tanto dados quanto processos. A fase de projeto orientado a objetos descreve como os objetos vão se comportar e interagir entre si. Objetos similares são agrupados para formar uma classe, e as classes são agrupadas em hierarquias nas quais uma subclasse herda os atributos e métodos de sua superclasse.

Para implementar o sistema de informação, traduz-se o projeto em um código-fonte de programa, reutilizando as classes que já estiverem disponíveis na biblioteca de objetos reutilizáveis e acrescentando novas classes criadas durante a fase de projeto orientado a objetos. A implementação também pode envolver a criação de um banco de dados orientado a objetos. O sistema resultante precisa ser exaustivamente testado e avaliado.

Como os objetos são reutilizáveis, o desenvolvimento orientado a objetos pode reduzir o tempo e o investimento necessários para escrever um software, caso as organizações reutilizem os objetos de software já criados como componentes de outras aplicações. É possível criar novos sistemas usando alguns objetos existentes, alterando outros e acrescentando alguns novos objetos.

### Serviços Web e desenvolvimento baseado em componentes

Para acelerar ainda mais a criação de softwares, grupos de objetos são reunidos na forma de componentes de software para funções comuns, tais como uma interface gráfica de usuário ou uma função de processamento de pedidos on-line, e esses componentes podem ser combinados para criar aplicações empresariais de grande escala. Essa abordagem de desenvolvimento de software é denominada **desenvolvimento baseado em componentes**. As empresas estão usando esse tipo de desenvolvimento para criar suas aplicações de comércio eletrônico; para tanto, elas combinam componentes disponíveis comercialmente — carrinho de compra, autenticação de usuário, mecanismos de busca e catálogos, por exemplo — com softwares específicos para seus requisitos empresariais.

No Capítulo 4, os *Web services* (serviços Web) foram definidos como componentes de software reutilizáveis, vagamente relacionados, baseados em *extensible markup language* (XML) e outros padrões e protocolos abertos que permitem que uma aplicação se comunique com outra sem a necessidade de programação customizada. Além de facilitar a integração interna e externa de sistemas, os serviços Web oferecem ferramentas não proprietárias para o desenvolvimento de novas aplicações de sistemas de informação, ou para o aprimoramento dos existentes. Os serviços Web executam certas funções por conta própria, e também podem utilizar outros serviços Web para completar transações mais complexas, tais como verificação de crédito, seleção de fornecedores e pedido de produtos. Por usarem um conjunto universal de padrões, esses serviços de software prometem ser menos caros e menos difíceis de interligar do que os componentes proprietários.

### Engenharia de software assistida por computador (CASE)

A **engenharia de software assistida por computador** (*computer-aided software engineering* — CASE) — às vezes também chamada de *engenharia de sistemas assistida por computador* — oferece ferramentas de software que automatizam as metodologias que acabamos de descrever, reduzindo a quantidade de trabalho repetitivo no

desenvolvimento de sistemas. As ferramentas CASE oferecem recursos gráficos automatizados para produzir gráficos e diagramas, dicionários de dados e geradores de telas e relatórios, além de contar com amplos recursos para a produção de relatórios, ferramentas de análise e verificação, geradores de códigos e de documentação. As ferramentas CASE também contêm recursos para validação de especificações e diagramas de projeto.

Tais ferramentas facilitam a documentação clara e a coordenação do trabalho da equipe de desenvolvimento. Os membros da equipe podem compartilhar seu trabalho acessando os arquivos uns dos outros para revisar ou modificar o que já foi feito. Se as ferramentas forem usadas adequadamente, também é possível obter um pequeno ganho de produtividade. Muitas ferramentas CASE são baseadas em PC, com recursos gráficos avançados.

## Resumo

**1 Identifique e descreva os principais passos de resolução de problemas para desenvolver novos sistemas de informação.** Os principais passos de resolução de problemas para o desenvolvimento de novos sistemas de informação são: (1) definir e compreender o problema, (2) desenvolver soluções alternativas, (3) avaliar e escolher a solução e (4) implementar a solução. O primeiro passo envolve definir o problema e identificar suas causas, os objetivos da solução e requisitos de informação. O segundo passo identifica soluções alternativas para o problema. O terceiro passo abrange a avaliação da viabilidade técnica, financeira e organizacional de cada alternativa e a seleção da melhor solução. O quarto passo abarca a finalização das especificações de projeto, a aquisição de hardware e software, testes, a oferta de treinamento e documentação, a conversão e a avaliação da solução de sistema uma vez que ela entre em produção.

**2 Avalie modelos para identificar o valor organizacional dos sistemas de informação.** Para determinar se um sistema de informação é um bom investimento para a empresa, é preciso calcular seus custos e benefícios. Os benefícios tangíveis são quantificáveis, e os intangíveis não podem ser imediatamente quantificados, mas podem proporcionar benefícios quantificáveis no futuro. Benefícios que superem os custos devem, então, ser analisados por meio de métodos de orçamento de capital, como o valor presente líquido (VPL), para se ter certeza de que representam um bom retorno sobre o capital investido da empresa.

Outros modelos para avaliar os investimentos em sistemas de informação envolvem considerações não financeiras e estratégicas. As organizações devem desenvolver planos de sistemas de informação que descrevam como a tecnologia de informação apóia o plano de negócios e a estratégia global da empresa. Análise de carteira (portfólio) e modelos de pontuação (*scoring*) podem ser usados para avaliar projetos e sistemas de informação alternativos.

**3 Estime os requisitos para gerenciar com sucesso a mudança criada pelos novos sistemas.** Uma imensa porcentagem de sistemas de informação não consegue proporcionar os benefícios ou solucionar os problemas como pretendido, porque o processo de mudança organizacional que cerca o desenvolvimento dos sistemas não foi adequadamente conduzido. As principais causas de fracasso de sistemas de informação são (1) participação inadequada ou insuficiente do usuário no processo de desenvolvimento do sistema, (2) falta de apoio da administração, (3) altos níveis de complexidade e risco e (4) gerenciamento de projeto deficiente.

O termo *implementação* refere-se a todo o processo de mudança organizacional que cerca a introdução de um novo sistema de informação. O projeto de sistemas de informação e todo o processo de implementação devem ser gerenciados como uma mudança organizacional planejada, por meio de uma análise de impacto organizacional. O apoio da administração e seu controle sobre o processo de implementação são essenciais, assim como mecanismos para lidar com o nível de risco em cada novo projeto de sistemas. Com o planejamento formal e as ferramentas de controle, é possível acompanhar as alocações de recursos e as atividades específicas de projeto. Os usuários podem ser estimulados a assumir papéis ativos no desenvolvimento de sistemas e envolver-se com a instalação e o treinamento.

**4 Avalie os diversos métodos para desenvolver sistemas de informação.** Existe uma série de métodos para o desenvolvimento de sistemas de informação, cada um adequado a diferentes tipos de problemas. O método mais antigo de desenvolvimento de sistemas é o ciclo de vida de sistemas, em que o desenvolvimento é feito em estágios formais. Os estágios devem evoluir em seqüência e ter resultados definidos. Cada um precisa ser formalmente aprovado antes que o próximo se inicie. O ciclo de vida de sistemas é rígido e dispendioso, mas adequado para grandes projetos que exijam especificações formais e rígido controle administrativo sobre cada estágio do desenvolvimento.

A prototipagem consiste em desenvolver um sistema experimental de maneira rápida e barata, para que os usuários finais interajam com ele e o avaliem. O protótipo é refinado e aperfeiçoado até que os usuários sintam que ele atende a todas as suas necessidades, podendo ser usado como modelo para criar o sistema final. A prototipagem estimula o envolvimento do usuário final no desenvolvimento de sistemas, assim como a iteração de projeto até que as especificações sejam capturadas com exatidão. Contudo, a criação rápida de protótipos pode resultar em sistemas não suficientemente testados e documentados, ou tecnicamente inadequados para o ambiente de produção.

Desenvolvimento pelo usuário final é o desenvolvimento de sistemas de informação feito por usuários finais, sozinhos ou com uma assistência mínima de especialistas em sistemas de informação. Os sistemas desenvolvidos dessa maneira podem

ser criados rápida e informalmente graças às ferramentas de software de quarta geração. As vantagens básicas de tais sistemas são a melhor determinação de requisitos; o menor acúmulo de tarefas; e a maior participação e controle do usuário final no processo de desenvolvimento de sistemas. No entanto, o desenvolvimento pelo usuário final, em combinação com a computação distribuída, vem introduzindo novos riscos organizacionais, com a propagação de sistemas de informação e recursos de dados que não necessariamente atendem a padrões de garantia de qualidade nem são facilmente controlados.

As empresas também podem desenvolver sistemas comprando um software ou serviços de software de vendedores externos. Uma alternativa é comprar um pacote de software aplicativo, que elimina a necessidade de escrever programas de software durante o desenvolvimento do sistema de informação. Os pacotes de software aplicativos são úteis se a empresa não conta com uma equipe de sistemas de informação ou com recursos financeiros suficientes para desenvolver um sistema sob medida.

É possível, ainda, terceirizar o desenvolvimento de sistemas. O outsourcing (terceirização) consiste em usar um fornecedor externo para desenvolver (ou operar) os sistemas de informação da empresa. O trabalho é feito por um fornecedor nacional ou estrangeiro, e não por uma equipe interna da empresa. Se for adequadamente gerenciado, o outsourcing pode poupar custos de desenvolvimento de aplicações ou permitir que as empresas desenvolvam aplicações sem uma equipe interna de sistemas de informação. No entanto, elas correm o risco de perder o controle sobre seus sistemas de informação e tornar-se dependentes demais de fornecedores externos.

Hoje, para permanecer competitivas, as empresas muitas vezes precisam desenvolver aplicações de e-commerce e e-business muito rapidamente. Por conta disso, recorrem ao desenvolvimento rápido de aplicações, ao projeto conjunto de aplicações (JAD) e aos componentes de software reutilizáveis (incluindo serviços Web) para acelerar o processo de desenvolvimento de sistemas.

**5 Compare as diversas metodologias para formatar e projetar sistemas.** As duas principais metodologias para formatação e projeto de sistemas de informação são as metodologias estruturadas e o desenvolvimento orientado a objetos. As metodologias estruturadas enfocam separadamente os dados e processos de formatação. O diagrama de fluxo de dados é a principal ferramenta para a análise estruturada, enquanto o diagrama de estrutura é a principal ferramenta para representar o projeto de software estruturado. No desenvolvimento orientado a objetos, o sistema é formatado como um conjunto de objetos que combinam processos e dados. O desenvolvimento orientado a objetos baseia-se nos conceitos de classe e herança.

## Termos-chave

Abordagem em fases, 345	Desenvolvimento pelo usuário final, 356	Implementação, 350
Agente de mudança, 350	Desenvolvimento rápido de aplicações (RAD), 360	Interface do usuário, 350
Análise de carteira (portfólio), 349	Diagrama de estrutura, 361	Manutenção, 345
Análise de impacto organizacional, 352	Diagrama de fluxo de dados, 361	Modelo de pontuação, 349
Análise de sistemas, 342	Dificuldade de comunicação usuário-projetista, 351	Plano de sistemas de informação, 348
Benefícios intangíveis, 347	Documentação, 345	Plano de teste, 345
Benefícios tangíveis, 347	Engenharia de software assistida por computador (CASE), 363	Produção, 345
Ciclo de vida de sistemas, 352	Ergonomia, 352	Projeto conjunto de aplicações (JAD), 360
Conversão, 345	Especificações de processo, 361	Projeto de sistemas, 344
Conversão direta, 345	Estruturados, 360	Prototipagem, 355
Conversão em paralelo, 345	Estudo de viabilidade, 343	Requisição formal de proposta, 358
Customização, 358	Ferramentas formais de planejamento e de controle, 352	Requisitos de informação, 342
Desenvolvimento baseado em componentes, 363		Teste, 345
Desenvolvimento orientado a objetos, 362		Teste de aceite, 345
		Teste de sistema, 345
		Teste de unidade, 345

## Questões de revisão

- 11.1** Enumere e descreva os passos de resolução de problemas para o desenvolvimento de um novo sistema.
- 11.2** O que são requisitos de informação? Por que eles são importantes para o desenvolvimento de uma solução de sistema?
- 11.3** Enumere os vários tipos de especificações de projeto necessários para um novo sistema de informação.
- 11.4** Por que a etapa de teste é tão importante no desenvolvimento de sistemas? Nomeie e descreva os três estágios de teste de um sistema de informação.
- 11.5** Que papel a documentação, a conversão, a produção e a manutenção desempenham no desenvolvimento de sistemas?
- 11.6** Qual a diferença entre benefícios tangíveis e intangíveis? Mencione seis benefícios tangíveis e seis intangíveis.
- 11.7** Quais são os principais componentes de um plano de sistemas de informação?
- 11.8** Descreva como a análise de carteira (portfólio) e os modelos de pontuação (*scoring*) podem ser usados para determinar o valor de sistemas.

**11.9** Por que é necessário compreender o conceito de implementação quando se gerencia a mudança organizacional que cerca um novo sistema de informação? Quais são os fatores que determinam o sucesso ou fracasso da implementação?

**11.10** O que significa dificuldade de comunicação usuário-projetista? Quais tipos de problema de implementação ela pode criar?

**11.11** Quais estratégias podem ser usadas para aumentar a probabilidade de sucesso de uma implementação de sistema?

**11.12** O que é o ciclo de vida de sistemas tradicional? Descreva suas vantagens e desvantagens.

**11.13** O que significa prototipagem de sistemas de informação? Quais são suas vantagens e limitações? Enumere e descreva as etapas no processo de prototipagem.

### Para discutir

**11.1** Discuta o papel dos usuários finais e dos profissionais de sistemas de informação no desenvolvimento de uma solução de sistema. Quando a solução é desenvolvida por meio do ciclo de vida de sistemas, da prototipagem ou do desenvolvimento pelo usuário final, como varia o papel de cada um desses grupos?

### Exercício com aplicativo

#### Exercício de banco de dados: projetando um sistema de cliente para venda de carros

Habilidades de software: projeto de banco de dados, consultas e relatórios

Habilidades organizacionais: gerenciamento de indicações e análise de cliente

A Ace Auto Dealers é uma concessionária especializada em veículos zero quilômetro da Subaru. Além de anunciar em jornais locais, a empresa está listada como concessionária autorizada no site da Subaru e em outros grandes sites para compradores de carros. A Ace desfruta de uma boa reputação local e de reconhecimento de marca, sendo uma das principais fontes de informações sobre veículos da Subaru na região de Portland, Oregon.

Quando um cliente potencial entra no showroom, é cumprimentado por um representante de vendas. O representante preenche manualmente um formulário com informações sobre o cliente potencial, tais como seu nome, endereço, número de telefone, dia da visita e o modelo e tipo de automóvel em que ele está interessado. Em seguida, o representante pergunta de que maneira a pessoa soube da Ace — se por um anúncio de jornal, pela Web ou por indicações —, e essa informação também é anotada no formulário. Se a pessoa decide comprar um carro, o vendedor preenche um termo de compra e venda.

A Ace acha que não possui informações suficientes sobre seus clientes. Ela não consegue determinar facilmente quais clientes potenciais acabam fazendo uma compra, ou a porcentagem de clientes potenciais que se convertem em compradores. Não consegue identificar, tampouco, quais pontos de contato com o cliente produzem o maior número de indicações de vendas ou de vendas efetivas, de maneira que possa focalizar a publicidade e o marketing nos canais que dão maior retorno. Será que os compradores estão descobrindo a Ace por meio de anúncios de jornal, de indicações ou da Web?

Prepare um relatório de análise de sistemas detalhando o problema da Ace e uma solução de sistema que possa ser implementada por meio de um software de gerenciamento de banco de dados para PCs. Use, então, um software de banco de dados para desenvolver uma solução de sistema simples. Seu relatório de análise de sistemas deve incluir os seguintes itens:

1. Descrição do problema e seu impacto gerencial e organizacional.
2. A solução proposta, seus objetivos e viabilidade.
3. Os custos e benefícios da solução que você escolheu. A empresa tem um PC com acesso à Internet e o pacote completo das ferramentas de produtividade Microsoft Office.
4. Requisitos de informação a serem atendidos pela solução.
5. Questões humanas, organizacionais e tecnológicas a serem consideradas pela solução, incluindo mudanças nos processos de negócios.

**11.14** O que é desenvolvimento pelo usuário final? Quais são suas vantagens e desvantagens?

**11.15** Quais são as vantagens e desvantagens do desenvolvimento de sistemas de informação a partir de pacotes de software aplicativos?

**11.16** O que é outsourcing? Sob quais circunstâncias ele deve ser usado para o desenvolvimento de sistemas de informação? Quais são os custos ocultos do outsourcing internacional de software?

**11.17** Como as empresas podem desenvolver rapidamente aplicações de e-business?

**11.18** Compare as abordagens estruturada e orientada a objetos na modelagem e projeto de sistemas.

**11.2** Já se disse que os sistemas fracassam quando os desenvolvedores de sistema ignoram os problemas 'humanos'. Qual seria o motivo?

Com base nos requisitos que você identificou, projete um banco de dados de clientes simples, alimentando-o com pelo menos dez registros. Use, então, o sistema que você criou para gerar consultas e relatórios de interesse da administração. Crie vários protótipos de formulários de entrada de dados para o sistema e examine-os com seu instrutor. Depois, refaça os protótipos.



### Projetando um sistema de treinamento e rastreamento de habilidades para funcionários

Habilidades de software: projeto de banco de dados, consultas e relatórios

Habilidades organizacionais: identificação de habilidades e treinamento de funcionários

A Dirt Bikes se define como uma 'empresa em constante aprendizado'; ela investe em cursos universitários ou de atualização para ajudar seus funcionários a progredir na carreira. Sua força de trabalho é bem jovem e volúvel. Quando algum funcionário deixa a empresa, seu cargo fica vago e a Dirt Bikes precisa preenchê-lo rapidamente para manter o ritmo de produção. A equipe de recursos humanos gostaria de encontrar uma maneira de identificar rapidamente funcionários com formação suficiente para preencher as vagas. Se a empresa souber quem são esses funcionários, haverá maior probabilidade de preencher os cargos abertos internamente, em vez de pagar pelo recrutamento de candidatos externos. A Dirt Bikes gostaria de identificar o nível de escolaridade de cada funcionário, bem como o nome e a data de conclusão dos cursos de atualização que ele frequentou.

Atualmente, a empresa não consegue identificar esses funcionários. Seu banco de dados de funcionários limita-se a dados básicos de RH, como nome do funcionário, número de identificação, data de nascimento, endereço, número de telefone, estado civil, cargo e salário. Você encontrará alguns exemplos de registros desse banco de dados, em inglês, no material do Companion Website para o Capítulo 11, que pode ser aberto com o Microsoft Access. A equipe de recursos humanos da Dirt Bikes mantém dados sobre o perfil e a formação dos funcionários em arquivos de papel.

Prepare um relatório de análise de sistemas descrevendo o problema da Dirt Bikes e uma solução de sistema que possa ser implementada por meio de um software de banco de dados para PCs. Use, então, o software de banco de dados para desenvolver uma solução de sistema simples. Seu relatório deve incluir os seguintes itens:

1. Descrição do problema e seu impacto gerencial e organizacional.
2. Solução proposta e seus objetivos.
3. Requisitos de informação a serem atendidos pela solução.
4. Questões humanas, organizacionais e tecnológicas a serem consideradas pela solução, incluindo mudanças e processos de negócios.

Com base nos requisitos que você identificou, projete um banco de dados de clientes simples, alimentando-o com pelo menos dez registros. Verifique se você pode usar o banco de dados de funcionários preexistente no seu projeto, ou incorporá-lo com modificações. Imprima o projeto de cada tabela da sua nova aplicação. Use o sistema que você desenvolveu para criar consultas e relatórios de interesse da administração (por exemplo, quais funcionários têm formação em gerenciamento de projetos ou ferramentas avançadas de projeto assistido por computador — CAD).

Se possível, use software de apresentação eletrônica para resumir suas conclusões à administração.

### Desenvolvendo habilidades de Internet

#### Analisando projetos de site e requisitos de informação

Este projeto ajudará você a desenvolver sua habilidade para analisar o projeto e a funcionalidade empresarial de sites.

Visite um site à sua escolha e explore-o profundamente. Prepare relatórios que analisem as várias funções do site e seus requisitos de informação. Seu relatório deve responder a estas perguntas: Quais funções esse site executa? Quais dados utiliza? Quais são as entradas, saídas e o processamento? Quais são algumas das suas outras especificações de projeto? O site se conecta a sistemas internos ou sistemas de outras organizações? Que valor esse site traz para a empresa?

### Trabalho em equipe

#### Analisando requisitos de site

Com um grupo de três ou quatro colegas, visite os sites iTunes, MP3.com e Internet Movie Database, ou o site de qualquer empresa descrita neste livro que use a Web. Examinem o site da empresa escolhida. Usem aquilo que vocês descobriram no site e o que aprenderam neste capítulo para preparar um relatório descrevendo as funções do site e algumas de suas especificações de projeto. Se possível, usem software de apresentação eletrônica para apresentar suas conclusões à classe.

## ESTUDO DE CASO: RESOLVENDO PROBLEMAS ORGANIZACIONAIS

### Como o sistema de escala de tripulação da Comair entrou em colapso

A Comair é uma companhia aérea regional sediada em Cincinnati, Ohio. A empresa emprega aproximadamente 7 mil profissionais de aviação, que, juntos, supervisionam e operam mais de 1.100 vôos diários com 30 mil passageiros. A Comair voa para 113 cidades nos Estados Unidos, no Canadá e nas Bahamas. A princípio uma operadora independente, é hoje uma subsidiária inteiramente controlada pela Delta Airlines. As suas rotas incluem as principais conexões e mercados dos Estados Unidos, como Atlanta, Cincinnati, Orlando, Nova York, Washington, D.C. e Boston. Em 2004, 12,6 milhões de passageiros voaram em pelo menos um dos 170 jatos regionais Bombardier CRJ da Comair.

Ao longo de sua trajetória, a empresa tem sido uma líder premiada no setor de aviação regional. A Administração Federal de Aviação (FAA) dos Estados Unidos e várias revistas de negócios já destacaram a Comair inúmeras vezes por sua lucratividade, sua gestão e suas estatísticas de pontualidade, cancelamento e perda de bagagem. As glórias do passado, porém, não foram de grande valia para a empresa durante as férias de inverno em dezembro de 2004, quando um sistema legado crítico falhou e a Comair sofreu um pesadelo de relações públicas — além, claro, de um duro golpe financeiro.

Em 25 de dezembro de 2004, um bug no software de escala da tripulação forçou a Comair a suspender todas as suas operações, mantendo em terra os 1.100 vôos já programados. A medida alterou e até mesmo arruinou os planos de férias de 30 mil passageiros. A Comair e a Delta perderam 20 milhões de dólares. Embora o catalisador do desastre de TI tenha sido uma infeliz seqüência de eventos climáticos adversos, a Comair atraiu ainda mais críticas ao usar o clima como principal justificativa da falha. Terry Tripler, especialista do setor que atua em Minneapolis, classificou a atitude da Comair como 'indesculpável' e comparou o desastre a uma situação em que todas as caixas registradoras do Wal-Mart quebrassem às vésperas do Natal.

A linha do tempo dos eventos que levam a dezembro de 2004 claramente demonstra que uma empresa pode crescer vigorosamente mesmo quando alguns de seus processos críticos são falhos. A Comair começou suas operações em 1977, com três aviões a hélice em Cincinnati. Em 1984, a Delta começou sua parceria com a pequena operadora regional. Graças a isso, a Comair se tornou um dos membros originais do programa de conexões da Delta.

Em 1984, quando a frota da Comair consistia em 25 aviões a hélice, a empresa gerenciava suas tripulações usando caneta e papel. Dez anos depois, as regulamentações federais e sindicais forçaram a Comair a adotar padrões mais elevados em seus procedimentos administrativos. Para atender às regulamentações, a empresa alugou um software da SBS International que controlava as tripulações, informando para quais vôos elas estavam designadas e quantas horas passariam no ar. O sistema cumpria suas tarefas de maneira admirável e, durante muitos anos, a Comair continuou sua marcha rumo ao topo do setor. Em 1993, ela se tornou a primeira no segmento a comprar um jato regional Bombardier CRJ. Mas essa vantagem durou pouco: em 1996, outras companhias aéreas regionais já haviam incorporado jatos à sua frota. A

Comair procurou, então, novas maneiras de ganhar vantagem competitiva.

Uma das áreas que a Comair queria aperfeiçoar eram seus sistemas de informação. Os sistemas da empresa executavam uma variedade de aplicativos para escala de tripulação, manutenção de aeronaves e reservas de assentos que, além de não se relacionarem entre si, estavam tornando-se obsoletos. Em 1997, o departamento de TI cogitou a substituição do sistema legado SBS que a empresa vinha usando para gerenciar a tripulação. Então com 11 anos de uso, o aplicativo tinha sido escrito em Fortran, uma linguagem de programação dos anos 1950, na qual ninguém do departamento era especializado. Era também o único aplicativo da empresa que ainda usava a antiga versão IBM AIX do sistema operacional UNIX, em vez do HP UNIX. Depois que a idéia foi debatida, a SBS visitou a Comair para tentar vender-lhe o seu mais recente software de gerenciamento de tripulação, o Maestro. No entanto, um dos supervisores de tripulação que assistiam à apresentação já conhecia esse produto de um emprego anterior e deu uma opinião desfavorável a respeito. Em vista disso, a Comair deixou o Maestro de lado e partiu em busca de uma solução melhor. Nesse meio tempo, os usuários finais da empresa trabalhavam de maneira razoavelmente eficiente com o sistema legado, e não tinham de ser retreinados para usar um sistema caro e de valor duvidoso.

Em 1998, Jim Dublikar, então diretor de gerenciamento de risco e tecnologia de informação da Comair, marcou uma consultoria com a SABRE Airline Solutions. Além de consultoria, a empresa de Southlake, no Texas, fornece também softwares para empresas aéreas. O objetivo era delinear uma estratégia de TI de longo prazo para administrar os sistemas legados e a infra-estrutura de TI da Comair. Cinco meses de reuniões produziram um plano quinquenal para avaliar a viabilidade dos sistemas existentes e, conforme necessário, eliminá-los, substituí-los ou aperfeiçoá-los. Um dos componentes-chave fadados à eliminação e substituição era o sistema de gerenciamento de tripulação. Dublikar achava fácil tomar essa decisão. Além de ser arriscado manter o antigo sistema em uso, novas tecnologias prometiam trazer benefícios financeiros graças ao aumento da produtividade e a um controle mais rígido de despesas. No entanto, a implementação dessa parte do plano foi prejudicada por uma série de incidentes.

No fim da década de 1990, o departamento de sistemas de informação da Comair dedicou a maior parte de seu tempo aos preparativos para o 'bug do milênio', o que envolvia programar os sistemas legados para reconhecer a mudança de século nas datas. Essa questão fazia parte do plano quinquenal, assim como lançar um sistema de passagens eletrônicas e um aplicativo de gerenciamento de receita, atualizar a rede corporativa e mudar o sistema de manutenção e engenharia. Em 1999, todas essas iniciativas estavam em curso ou finalizadas. Substituir o sistema de gerenciamento de tripulação era o passo seguinte. As coisas nessa área avançavam lentamente, pois a empresa estava acostumada demais ao sistema SBS, e muitos de seus processos e regras de negócios (tais como a definição da jornada de um piloto) estavam relacionados ao antigo software. A escolha de um novo sistema

de gerenciamento de tripulação foi finalmente agendada para o ano 2000. No entanto, antes disso, a Comair atravessou um período de turbulência. Primeiro, Dublikar deixou a organização. Depois, a Delta comprou a empresa, o que trouxe novas vozes para o processo de tomada de decisão.

A Delta viu a Comair como uma aquisição fácil. Sua antiga parceira apresentava lucros de maneira consistente, estava saindo-se bem na bolsa de valores e era líder nas estatísticas de desempenho. Com base nessa perspectiva, a Delta não via razões para mexer nas operações da Comair. O foco da Delta não era TI, e sim marketing. De fato, a Delta instalou seu próprio departamento de marketing na Comair logo depois da aquisição.

Dublikar só foi substituído no início de 2000, o que deixou um vácuo de liderança no departamento de TI. De acordo com um ex-funcionário de TI da empresa, Eric Bardes, a equipe não queria criar polêmica em torno da aquisição e da falta de um líder. Eles se limitaram a esperar que a parte administrativa da empresa estimulasse suas iniciativas. Enquanto isso, a parte administrativa esperava que o departamento de TI fosse proativo. E, assim, os projetos remanescentes do plano quinquenal, incluindo a substituição do sistema de gerenciamento de tripulação, ficaram pelo caminho.

Em 2001, outro evento tirou o foco de atenção da Comair dos seus projetos de TI. Uma greve de pilotos com duração de 89 dias quebrou as pernas tanto da companhia aérea quanto do Aeroporto Internacional Cincinnati / Northern Kentucky, onde a Comair opera 90 por cento de seus vôos. Com 800 vôos diários cancelados, a Delta perdeu 200 milhões de dólares naquele trimestre. Em junho, quando a greve foi suspensa, a equipe de operações de vôo não podia simplesmente apertar um botão mágico e retomar as atividades. A trabalhadora de programar vôos e tripulações impedia a empresa de pensar na substituição do sistema de escala.

Poucos meses depois, a instabilidade causada pela greve pareceu pequena perto do impacto dos ataques terroristas em 11 de setembro de 2001. A Delta sofreu prejuízos de quase 9 bilhões de dólares nos quatro anos seguintes. No fim de 2002, o departamento de TI da Comair convidou a SABRE, a SBS e outros fornecedores para demonstrar seus sistemas de gerenciamento de tripulação. Mas a empresa não fechou nenhum contrato naquele momento por questões de custo. Somente em junho de 2004 a Delta finalmente aprovou a substituição do sistema legado de gerenciamento de tripulação. A Comair decidiu-se pelo AirCrews Operations Manager, da SABRE, e agendou o lançamento para 2005. Evidentemente, 2005 não chegou rápido o suficiente.

No fim de dezembro de 2004, uma rigorosa tempestade de inverno caiu sobre o Vale Ohio. A neve e o gelo prejudicaram de tal maneira os aviões, pistas e operações que a Comair teve de cancelar ou atrasar mais de 90 por cento de seus vôos entre 22 e 24 de dezembro. O mau tempo e os

cancelamentos seriam, contudo, apenas parte de um problema muito maior. No dia 25, o software de gerenciamento de tripulação, então com aproximadamente duas décadas de uso, simplesmente entrou em colapso. Ninguém da Comair sabia que o software registrava mudanças de escala com um contador antiquado, que não comportava mais de 32.768 mudanças por mês. As tempestades dos dias anteriores haviam levado a tantas mudanças de escala que o software chegou ao limite e entrou em pane. Toda a programação de vôos de 25 de dezembro e 90 por cento da programação do dia seguinte simplesmente desapareceram. A Comair não tinha um sistema reserva e o fornecedor de software precisou de um dia inteiro para reverter a falha. A empresa só voltou a funcionar normalmente em 29 de dezembro, mas o estrago já estava feito. Em questão de alguns poucos dias, a Delta perdeu praticamente todo o lucro produzido pela Comair no trimestre anterior.

A Comair sustenta que a peça central do problema foi o mau tempo, não as limitações de seu velho software. Em março de 2005, a companhia aérea ainda não havia implementado o novo pacote de software SABRE. Continuava usando o sistema legado SBS, agora dividido em dois módulos, de maneira que mudanças na escala dos pilotos e mudanças na escala das aeromoças tenham, cada grupo, um limite mensal de 32 mil. A Comair também está monitorando o volume de transações com mais cuidado.

Fontes: Stephanie Overby, "Bound to Fail", *CIO Magazine*, 01 maio 2005; [www.comair.com](http://www.comair.com), acessado em 30 jun. 2005; [www.sabreairlinesolutions.com](http://www.sabreairlinesolutions.com), acessado em 30 jun. 2005; TechWeb News, "Comair downed by computer counting limit", *TechWeb*, <http://www.techweb.com/wire/56700130>, acessado em 29 dez. 2004; Katie Fairbank, "Airlines holiday troubles blasted", *Dallas Morning News*, conforme publicado em [www.jsonline.com/bym/news/dec04/287755.asp](http://www.jsonline.com/bym/news/dec04/287755.asp), acessado em 27 dez. 2004; Jim Wagner, "Comair back in air after computer outage", *IT Management*, <http://itmanagement.earthweb.com/erp/article.php/3451981>, acessado em 27 dez. 2004; e Jim Wagner, "Feds to probe Comair after computer outage", *IT Management*, <http://itmanagement.earthweb.com/erp/article.php/3452501>, acessado em 28 dez. 2004.

#### Questões do estudo de caso

1. Quais eram os problemas enfrentados pela Comair com seus sistemas? O que causava o problema? Qual era o seu impacto?
2. Quais eram as soluções disponíveis à administração? A administração escolheu a melhor solução? Por quê? Você consegue pensar em outras soluções que a Comair poderia ter considerado?
3. Quais aspectos humanos, organizacionais e tecnológicos influenciaram os processos de tomada de decisão das partes envolvidas neste caso?
4. Quais questões humanas, organizacionais e tecnológicas a Comair tem de considerar se resolver desenvolver um novo sistema?