Chapter

13

Business Process Management and Systems Development

Learning Objectives

❶ Understand business process management (BPM), BPM tools, and service-oriented architecture (SOA)—and their role in business agility and process optimization.

❷ Understand the importance of software architecture design to the maintenance and agility of business processes.

❸ Describe IT project identification, justification, and planning; understand the triple constraints.

❹ Describe the systems development lifecycle (SDLC).

Integrating *IT*

ACC	FIN	MKT	OM	HRM	IS

CHAPTER 13 LINK LIBRARY

ARIS Express, free business process modeling software. *ariscommunity.com/aris-express*
Oracle BPM Suite 11g *oracle.com/us/technologies/bpm/*
Oracle SOA Suite 11g *oracle.com/us/technologies/soa/*
Project Management Institute *pmi.org/*
Fastforward BPM blog *fastforwardblog.com/2010/06/26/social-bpm-business-process-management-enters-the-21st-century/*
Open source BPM and workflow *processmaker.com/*
BPM/SOA Community Insights *blog.soa-consortium.org/*
Adaptive Planning demo for budgeting, forecasting, reporting, and analysis *adaptiveplanning.com/*
IBM BPM *www-01.ibm.com/software/info/bpm/*
IBM BPM Blueprint demo (download the demo) *www-01.ibm.com/software/integration/bpm-blueprint/*
ITBusinessEdge BPM *itbusinessedge.com/topics/show.aspx?t=482*
Oracle Business Activity Monitoring (BAM), integral part of the BPM suite *oracle.com/appserver/business-activity-monitoring.html*
InfoSys Research BPM, SOA, and enterprise architecture; Centers of Excellence
infosys.com/research/
infosys.com/research/centers-of-excellence/

QUICK LOOK at Chapter 13, Business Process Management and Systems Development

This section introduces you to the business issues, challenges, and IT solutions in Chapter 13. Topics and issues mentioned in the Quick Look are explained in the chapter.

Ever-changing ITs, company mergers, industry consolidations, regulatory requirements, financial conditions, customer expectations, and global competition—what are these forces doing to organizations and the business climate? The simple and obvious answer is that they are causing rapid changes. Being able to redesign business processes to respond to those changes can be extremely complex, as you read in this chapter.

Business leaders know that each type of change—whether in the form of an opportunity or a threat—demands a smart (informed) response. Those demands trickle down to the business process level—the building blocks of each functional area. A **business process** is any system or procedure that an organization uses to achieve a larger business goal. Examples of business processes are:

- Accounting business processes
  - Accounts receivable (A/R) and accounts payable (A/P)
  - Bank account reconciliations
  - Cash receipts
- Finance business processes
  - Business forecasting
  - Financial cash flow reports
  - Credit approval and terms
- Human resources (HR) business processes
  - Employee hiring, screening, and training
  - Occupational health and workplace safety
  - Payroll
- Marketing business processes
  - Sales forecasting
  - Media campaigns
  - Customer service
- Management information systems (MIS) business processes
  - Network design and implementation
  - Data management
  - Information security and incident response
- Production and operations management business processes
  - Product design and development

- Quality control and assurance
- Shipping, receiving, and inventory management

Common to all business processes is that they *change*—and the management of those changes is the key topic of this chapter: business process management (BPM). To manage and redesign processes successfully, companies need a sound BPM strategy and the right set of tools.

Closely related to, but distinct from, BPM is **service-oriented architecture** (SOA). BPM is about modeling, implementing, and monitoring business processes; and most business processes entail several functions and/or services. SOA is a technology approach to implementing a business process, but it's only part of the technology required to implement business processes. Also in this chapter, we discuss IT project management—a disciplined approach to developing systems that meet specifications and are completed on time and within budget. Then you learn about the systems development process.

Microsoft International's HR Team Optimizes Business Processes

GLOBAL HRM IS

Microsoft International provides sales, marketing, and services for Microsoft Corporation's locations outside of North America. The human resources (HR) team within Microsoft International is made up of approximately 600 employees and provides support for HR management in more than 100 countries. HR performs many legal and staffing functions; key among them are recruiting, training, and employee development as well as compliance with regulations and health and safety laws, such as the U.K. Employment Law, U.K. health and safety regulations (*direct.gov.uk/*), and OSHA (Occupational Safety and Health Administration, *osha.gov/*) regulations in the United States. HR functions also include managing employee benefits, compensation, employee records, and personnel policies. Policies are often set forth in employee manuals, which are posted on the companies' intranets.

Microsoft's HR Business Processes

The HR team uses many global systems and tools across each of the international Microsoft subsidiaries. One key HR objective was to standardize common business processes of all subsidiaries across all Microsoft locations. As each subsidiary developed its own unique business processes, such as training new hires, there was no standardized way to compare, manage, or evaluate the efficiency or effectiveness of the business processes. (As you have read, *you can't manage what you can't measure*.) HR believed that the costs and time required to perform its common activities and train new employees were much higher than they could be. Jean O'Connor, HR project manager for Microsoft International, explained:

> *Experience with different HR business processes would vary significantly from one location to the next. Without documentation, each new HR employee will need to be trained by someone who may or may not know the process. Teaching new hires an inefficient process can introduce repeatable errors and decreases our overall effectiveness.*

HR Inefficiencies

The lack of standardized business processes and process documentation had a number of adverse impacts on the HR team:

- Increased the time and cost to train new employees because there was no simple way to describe critical HR processes
- Limited ability to review its business processes and make informed decision regarding the sequencing of steps and roles and responsibilities involved
- Decreased business process efficiency, with wide ranges in time to complete tasks across subsidiaries

Increasing HR Processes Efficiency

The team wanted to find ways to improve process efficiency and effectiveness across subsidiaries. To start, the HR team needed to understand current business processes at each subsidiary and be able to discuss them, which they achieved by diagramming them using Visio modeling software (*visiotoolbox.com/2010/home.aspx*). An example of a diagram of a business process is shown in Figure 13.1. (Visit the Microsoft Visio 2010 Web site at *visiotoolbox.com/2010/home.aspx* for more examples of business process modeling.)

After the workflows and information processing involved in a business process are accurately mapped out using standard notation, that process is ready to be analyzed so as to identify how to improve it. Equally important, these maps (or models) provide the starting point for standardizing the language used to describe their tasks.

Benefits of Business Process Modeling

The HR team used Microsoft Visio Premium 2010—a business process modeling tool—to design templates (also called models) that define and describe the steps in each process. The templates help HR staff understand Microsoft's standardized processes and are used in staff training. Benefits that the HR teams achieved are:

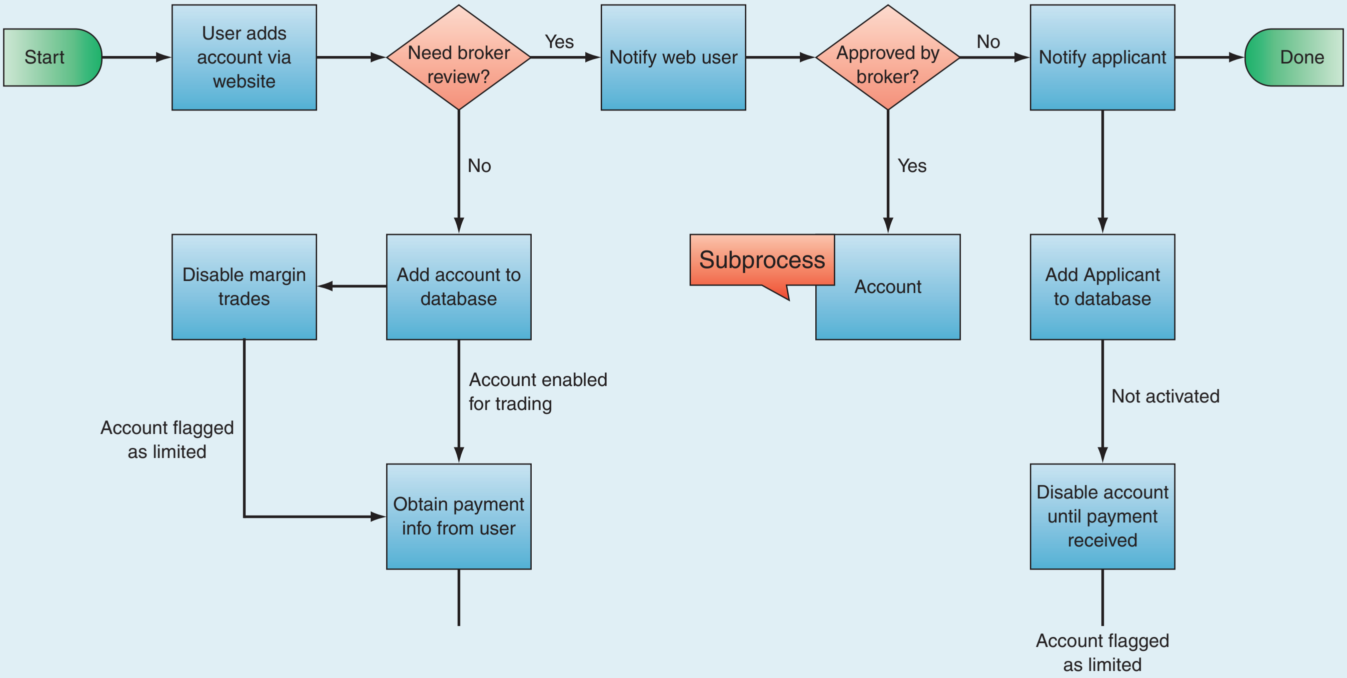

Start
User adds account via website
Need broker review?
Yes
Notify web user
Approved by broker?
No
Notify applicant
Done
No
Yes
Disable margin trades
Add account to database
Subprocess
Account
Add Applicant to database
Account enabled for trading
Account flagged as limited
Not activated
Obtain payment info from user
Disable account until payment received
Account flagged as limited

Figure 13.1 Model of key activities for setting up a customer account.

- **Significant savings in labor hours through increased process efficiency.** The models significantly reduced the time required to execute HR processes in all subsidiaries. According to O'Connor, "The key benefit for the HR organization is increased productivity through the creation of standardized process documentation across all of our sales, marketing, and services processes." The increased standardization helps clarify roles and responsibilities across HR employees and reduces the time HR teams spend on administrative tasks. As a result, "the right people are doing the right work at the right time across our business processes," said O'Connor.
- **Decrease in the training time of newly hired employees.** As HR processes are standardized from one subsidiary to the next, it's easier for an HR member from one country to move to another because roles, responsibilities, and process steps are similar in all locations. This further reduces the time and cost associated with training.
- **Improved decision making through visual process analysis.** Visual displays make it easier to understand and communicate about processes than using text.

Sources: Compiled from *Microsoft.com* (2010), Visio 2010, and *visimation.com/.*

For Class Discussion and Debate

1. ***Scenario for Brainstorming and Discussion:*** Why did Microsoft International have inefficient HR business processes until this HR initiative was completed in 2010? Does it seem strange that a mega-multinational computer software and services company had not been able to ensure that "the right people are doing the right work at the right time across our business processes" until 2010? What may have motivated or pressured Microsoft International to standardize its HR processes?

2. ***Debate:*** Select a process that you are familiar with, such as withdrawing cash from an ATM, registering for courses for the semester, or applying for a new job. Selecting a business process that everyone understands would be best. Working individually or in small teams of two or three, diagram all of the tasks needed to perform the process, making sure that you show the physical flows and the information flows in your model. After everyone or every team has completed their model diagrams, debate the tasks and flows until you have agreed to a standard (single) model to represent the process. Then continue to work on the model to improve the efficiency of the process. (You can expect to learn how difficult it is to people to agree on how a complex process is performed and how to improve it. You learn that modeling a business process is a series of debates over disagreements until agreement is reached.)

13.1 Business Process Management (BPM) and Service-Oriented Architecture (SOA)

BUSINESS PROCESSES AND TASKS

A **business process** accomplishes or produces something of value to the organization. A business process consists of a collection of tasks or activities that are executed according to certain rules and with respect to certain goals. For example, the credit approval process follows rules that take into consideration credit scores, debt, and annual salary to estimate the borrower's risk. The goal is to extend credit to those who are below some risk level.

When you break it down, you see that a business process is actually a series of individual tasks, and each task is executed in a specific order. A **task** is the smallest unit of work and management accountability that is not split into more detailed steps. The order of tasks/activities may be fully or only vaguely defined. Tasks can be automated, semiautomated, or performed manually.

A process has inputs and outputs that are *measurable* and therefore can be managed. Most processes cut across functional areas. For example, a product development process cuts across marketing, research and development, production, and finance (product development needs to be financed). Business processes are becoming more and more complex—composed of interactions across systems and dependent on collaborative activities between business users and IT. Complex processes often need to be broken into a number of subprocesses for easier management. When processes are designed for maximum efficiency, they said to be *optimized.*

Business Process Life Cycle. Business processes integrate ISs and people. Purchase order processing, staff recruitment, patient billing, order fulfillment, and everything else an organization does consist of processes that are performed by employees using ISs. Management of business processes boils down to the management of their life cycles, as shown in Figure 13.2. Business processes are introduced, modified to the extent possible, and get replaced—the standard format of a *life cycle.* Changes may require only simple adjustments to the tasks or rules of the process, such as changing the sales commission percent, or may be reengineered, such as changing the HR function, as you read in the Microsoft International opening case.

Design Stage. The cycle starts with process design. Process design is typically mapped and documented using a modeling tool, such as IBM BPM Blueprint or Microsoft Visio. This model plays a key role and, once finalized, serves as documentation of the entire process.

During the design stage, the team of business analysts and technology experts brainstorm possible solutions to current problem areas or opportunities. The design and functional specifications (specs) are completed at this phase. The *design spec*, also called the *technical spec*, identifies how the business process will be implemented in as much detail as possible. This spec identifies which systems are involved in the process, how they integrate, and the technical details of the implementation.

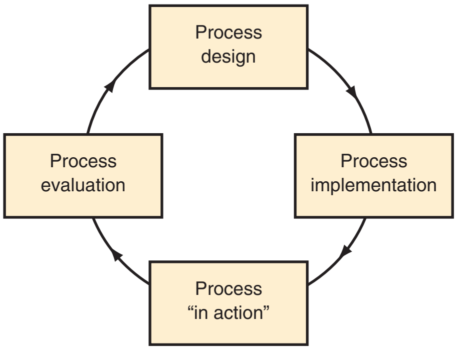


Figure 13.2 Business process life cycle.

Functional and technical specs can be hundreds of page long, which explains why specialized modeling tools are essential. The deliverables from the design stage are not all technical. The design spec also identifies how process users interact and complete tasks.

Implementation Stage. The business process agreed to in the design stage is delivered. Implementation includes integrating the process with other processes that share inputs or outputs, testing, and verifying that it works correctly and reliably. Problems may require going back to the process design stage.

Not only is the development of the process important, the testing of the process is equally as critical. The three tests are:

- **User acceptance:** Users tests whether the process is designed well from their perspective.
- **Functional acceptance:** Process analysts test whether the process performs its functions.
- **System acceptance:** Technical experts attest that the process is integrated correctly with inputs and outputs of other processes, data sources, and data stores.

After tests and refinements are completed, the process is ready to *go live,* and be put into action. After being put into action, the process is said to be "in production."

Process "In Production" and Evaluation Stages. As new processes are added or processes are redesigned or removed, processes that are in production may become problematic or unstable. Therefore, during this stage, the processes are monitored. Many software vendors that implement business processes, such as Oracle, Microsoft, Cordys, and IBM, include tools with **business activity monitoring (BAM)** functionality. For example, Oracle BAM is an integral part of the BPM suite (*oracle.com/appserver/business-activity-monitoring.html*). It is a message-based, event-driven tool that allows business users to link KPIs (key performance indicators) associated to the process being monitored on a real-time basis and provides relevant information via dashboards.

BUSINESS PROCESS MANAGEMENT

Business process management (BPM) is a fundamental management technique that includes methods and tools to support the stages of the business process life cycle. In the short term, BPM helps companies improve profitability by reducing waste and costs; in the long run, BPM helps keep companies responsive to business changes.

The BPM approach has its roots in **business process reengineering (BPR).** BPR is the radical redesign of an organization's business processes. BPR first attempts to *eliminate* processes that no longer have any purpose, often because of new mobile apps, Web services, or other IT. The processes that remain are redesigned and automated to the extent possible.

BPR quickly became a management fad, similar to just-in-time (JIT) inventory management. BPR and JIT were both based on assumptions. And if those assumptions were not met, then they failed to achieve the great expected results. That is, BPR was not understood enough and was applied incorrectly, with terrible results. Many JIT implementations increased inventory costs because JIT was based on the assumption that warehousing costs were extremely high, as they were in Japan, where JIT was initiated by Toyota. Why? Because JIT increases transportation and ordering costs. The increase in the costs must be offset by an even larger drop in warehousing costs. If not, JIT is more expensive. With BPR, companies first have to analyze and understand the inefficiencies in their business processes. They then have to figure out how to drive out waste and streamline processes and design those processes to minimize the risk of errors that led to rework. Then, and only then, should remaining processes be designed and automated. Many companies skipped the beginning steps and jumped to downsizing—firing employees. A manager at one of the major telecoms, in a discussion with one of the authors, lamented that "we amputated before we diagnosed." In addition to business disruptions, labor costs increased sharply as companies rehired employees. Therefore, in the 1990s most organizations failed to achieve fundamental process improvement because they attended a BPR seminar and then made mistakes in the implementation.

Despite decades of reengineering attempts, organizations still have problems with their business operations. They duplicate processes. They perform hundreds of noncore tasks that should be outsourced, and they spend vast amounts on proprietary process-management software that's difficult to update. To address these issues, BPM has evolved as a technique that ties people, processes, and technology to strategic performance improvement goals. To properly address process improvement, organizations must develop a carefully crafted BPM strategy.

BPM Strategy Considerations. Specifically, a well-implemented BPM strategy enables an organization to

- Gain greater visibility into processes
- Identify root causes of bottlenecks within processes
- Pinpoint the time and conditions when data from a process is handed-off (transferred) to other processes.

Done correctly, BPM helps an organization cut costs, improve service, achieve growth, or comply with regulations. For example, a manufacturer with a strategic goal of improving product quality and reliability must look at its manufacturing processes and see how they link to this business objective. If organizations focus exclusively on automation and cost savings, they might achieve significant operational efficiencies but lose their competitive edge and fall short of their performance targets, as British Telecom (BT) and United Airlines did when they failed to link strategic goals with their BPM initiatives.

Once the assessment is complete, it is necessary to develop a process performance plan that documents the ways in which the identified operational processes contribute to strategic goals. If a strategic goal is customer satisfaction, for example, appropriate process benchmarks should be established to accurately and consistently analyze progress of a BPM initiative. In improving an order-to-fulfillment process, although order throughput and on-time delivery are important, other measures might have a direct impact on customer satisfaction, such as fulfillment accuracy.

Finally, processes must be prioritized, with highest priority given to those processes that are determined to have the greatest potential impact on strategic objectives.

SERVICE-ORIENTED ARCHITECTURE (SOA)

SOA is a confusing concept, even for practitioners, for one of two reasons—either because SOA is mistakenly described like BPM or the definition of SOA is incomprehensible. To illustrate the latter, this is how IBM defines SOA on its Web site *http://www-01.ibm.com/software/solutions/soa/:*

> *Service Oriented Architecture (SOA) is a business-centric IT architectural approach that supports integrating your business as linked, repeatable business tasks, or services. With the Smart SOA approach, you can find value at every stage of the SOA continuum, from departmental projects to enterprise-wide initiatives.*

It's as if someone in the legal department wrote that IT definition in language few others could understand. Another definition of SOA was: "SOA is essentially a collection of services." Clearly that definition does not help either.

Services are like reusable software programs, or modules. You might even compare them to a macro in Excel. You can use and reuse them instead of writing code to perform common functions.

Oracle offers a technical explanation of SOA, which you find in Table 13.1. The *Public Sector Case* at the end of this chapter shows the value of SOA.

TABLE 13.1 SOA Defined

SOA is an architectural style for building software applications that use services available in a network such as the Web. It promotes loose coupling between software components so that they can be reused. Applications in SOA are built based on services. A service is an implementation of well-defined business functionality, and such services can then be used by clients in different applications or business processes.

SOA allows for the reuse of existing assets where new services can be created from an existing IT infrastructure of systems. In other words, it enables businesses to leverage existing investments by allowing them to reuse existing applications, and promises interoperability between heterogeneous applications and technologies. SOA provides a level of flexibility that wasn't possible before in the sense that:

- Services are software components with well-defined interfaces that are implementation-independent. An important aspect of SOA is the separation of the service interface (the what) from its implementation (the how). Such services are consumed by clients that are not concerned with how these services will execute their requests.
- Services are self-contained (perform predetermined tasks) and loosely coupled (for independence).
- Services can be dynamically discovered.
- Composite services can be built from aggregates of other services.

See *java.sun.com/developer/technicalArticles/WebServices/soa/*

BPM and SOA: Business Optimization. BPM and SOA are two of the most talked-about business initiatives. Both promise to help companies create new value from existing IT investments. They reuse IT programming efforts (think macros or modules) across many other processes. They also promise to enable agility through greater flexibility and lower cost structures.

The two are often confused because they confer many of the same benefits. SOA focuses on creating a more flexible IT architecture, while BPM has a pure focus on optimizing the way actual work gets done. SOA has delivered business value to very large corporations, but almost all SOAs in practice are used only in Web services, application integration as middleware, and B2B solutions.

BPM Mashups Through Web Services. Business processes are not self-contained. They need information from people and ISs (data stores) across departments and business areas. Many business processes even require information to be shared with external partners, clients, and providers. Web services can expand the functionality of the BPM system. A **Web service** is a set of technologies used for exchanging data between applications. Web services can connect processes with other systems across the organization and with business partners. The resulting integrated BPM systems are called **BPM mashups.**

Mashups are preconfigured, ready-to-go integrations between different business software packages. They streamline information sharing among systems. For example, a BPM system can leverage Web services to share customer data with CRM (customer relationship management). Budget and cost data from an ERP (enterprise resource planning) can be shared with the BPM, either to approve or deny an expense report filed using the BPM and subsequently to update the ERP once the expense report is complete. Web services can be used to share information with any other system that uses Web services. Mashups make the sharing process easier by providing the systems integration and streamlining the way that the two systems work together.

Review Questions

1. What is a business process? Give three examples.
2. What are the stages in the business process life cycle?
3. Define *business process management.*
4. Why is BPM important?
5. What is a BPM mashup?

13.2 Software Architecture and IS Design

An organization's software architecture refers to the structure of its applications. As with roads and bridges, architecture determines what is possible and the ease with which changes can be made to systems and processes.

AN OVERVIEW OF COUPLING IN SOFTWARE APPS

Long ago, business applications were written in COBOL software. These apps were one large piece or tightly coupled programs that performed many functions. *Tightly coupled* means that the programs and the data they processed and reports they generated were hardwired. Changes to these apps were tedious and time-consuming, as the Y2K problem demonstrated. For an explanation of the Y2K, or millennium bug, see *cybergeo.com/y2k/fulldetails.*html or search for online articles.

The preferred software design is loosely coupled and performs a single function or very few functions. What does *loose coupling* mean?

Loosely Coupled. Loose coupling refers to the way in which components in a system or network are connected. Loosely connected components have minimal dependence on one another. This simplifies testing, maintenance, and troubleshooting procedures because problems are easy to isolate and unlikely to spread or propagate. The extent, or "tightness," to which the components in a system are coupled is a relative term. A loosely coupled system can be easily broken down into definable elements.

The goal of loose coupling is to reduce dependencies between systems. Benefits of loose coupling include flexibility and agility. A loosely coupled approach offers unparalleled flexibility for adaptations to changing landscapes. Since there are no assumptions about the landscape your application is running against, you can easily adapt the composite application as needed.

Another aspect to consider is the probability of landscape changes during the lifetime of the application. Due to mergers and acquisitions and system consolidations, the landscape underneath the application is constantly changing. Without loose coupling, organizations are forced to adapt or rewrite their apps again and again.

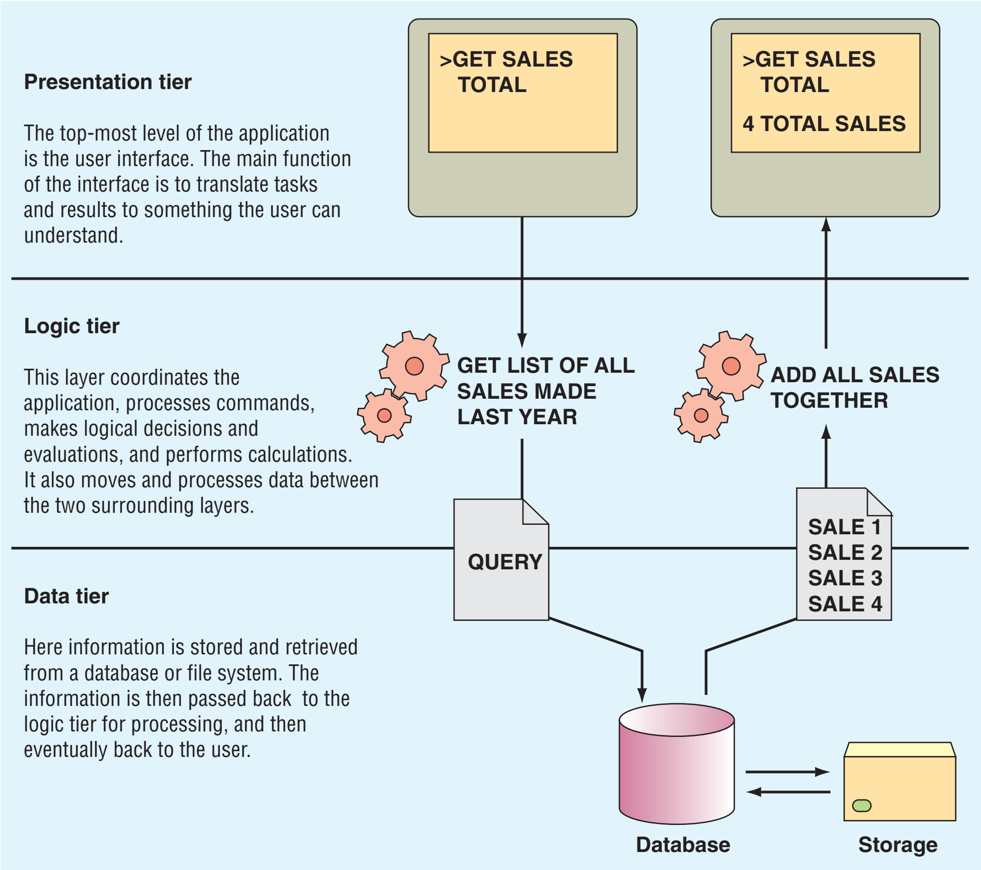
Maximizing Architecture Flexibility. An organization's software architecture can also be designed for greater flexibility by using a tiered model. An example of a three-tier architecture model is shown in Figure 13.3.

Notice the modular architecture. The three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as business requirements or technologies change. For example, a change of OS (operating system) in the presentation tier would only affect the user interface code.

Typically, the user interface runs on a PC, laptop, or handheld and displays a standard graphical user interface (GUI). The middle tier does the processing and coordinating of the data. The middle tier may be multitiered itself, which is called an n-tier architecture.

Three-tier architecture has the following three tiers:

1. **Presentation or client tier.** This is the topmost level of the application, an example of which is your Web browser. The presentation tier displays information related to such services as browsing merchandise, purchasing, and showing shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.
2. **Application or business logic tier.** Detailed processing is performed in this tier. This middle tier consists of **middleware.** Middleware refers to a broad range of software or services that enable communication or data exchange between applications across networks. Specifically, middleware enables the data exchange by translating data requests and responses between clients and servers. This type of software is often described as "glue" because it connects or integrates business-critical software



Presentation tier

The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.

Logic tier

This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.

Data tier

Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.

Figure 13.3 Overview of a three-tier software architecture design. (*Courtesy of Bartledan (Wikipedia, 2009).*)

applications to other applications. By performing some of the tasks that an application would have performed, middleware eliminates the need for an application that is shared by multiple clients to function differently for each different type of client.

With today's network-based applications—especially ERP, SCM, CRM, B2B, and B2C e-commerce—business operations depend on middleware to provide secure data transfers between these applications.

3. Data tier. This tier consists of the data sources, such as the database and data warehouse servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.

Conceptually the three-tier architecture is linear. A fundamental rule in a three-tier architecture is that the presentation tier never communicates directly with the data tier; all communication must pass through the middleware tier.

With this understanding of tiered architecture, we now discuss the IT acquisition process. Recall from Chapter 12, which focused on outsourcing strategies, that developing ISs in-house was the alternative option. We examine that option for developing business processes next.

THE IT ACQUISITION PROCESS

The acquisition process of an IT application has five major steps, which are shown in Figure 13.4 and discussed next.

Step 1: Planning, identifying, and justifying IT-based systems. IT systems are usually built as *enablers* of some business processes. Therefore, their planning must be aligned with the organization's overall business plan and the specific tasks they intend to support. Often processes may need to be redesigned or restructured to fully reap the benefits of the supporting IT applications. Also, the systems may need to be justified, for example, by cost-benefit analysis. Both of these activities may be complex,

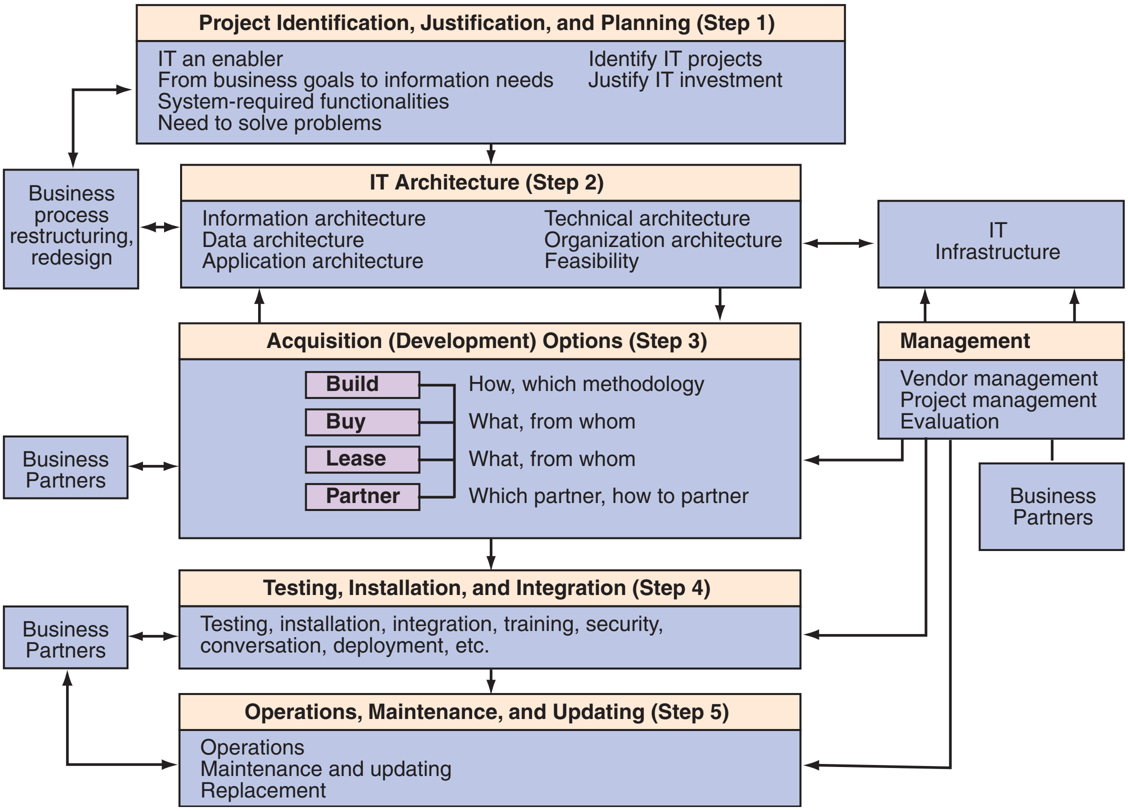

Project Identification, Justification, and Planning (Step 1)
IT an enabler
From business goals to information needs
System-required functionalities
Need to solve problems
Identify IT projects
Justify IT investment
Business process restructuring, redesign
IT Architecture (Step 2)
Information architecture
Data architecture
Application architecture
Technical architecture
Organization architecture
Feasibility
IT Infrastructure
Acquisition (Development) Options (Step 3)
Build
Buy
Lease
Partner
How, which methodology
What, from whom
What, from whom
Which partner, how to partner
Management
Vendor management
Project management
Evaluation
Business Partners
Business Partners
Testing, Installation, and Integration (Step 4)
Testing, installation, integration, training, security, conversation, deployment, etc.
Business Partners
Operations, Maintenance, and Updating (Step 5)
Operations
Maintenance and updating
Replacement

Figure 13.4 The process of IT application acquisition.

especially for systems that require a significant investment to acquire, operate, and maintain—or that are cutting-edge.

The output of this step is the decision to invest or not invest in a specific application and a timetable, budget, and assigned responsibility. This step is usually done in-house, with consultants as needed. All other steps can be done in-house or outsourced.

The importance of a realistic evaluation cannot be overstated. Many projects pass this stage because of political reasons or fear of taking an unpopular position. Managers may hope that the system will work out. *Hope is not a plan*—it's a risk. *IT at Work 13.1* describes the multibillion-dollar failure of the plan by the U.S. Census to collect data using handheld devices.

Step 2: Creating IT architecture. IT architecture is a plan for organizing the underlying infrastructure and applications of the IT project. The architecture plan includes the following:

- Data required to fulfill the business goals and vision
- Application modules that will deliver and manage the information and data
- Specific hardware and software on which the application modules will run
- Security, scalability, and reliability required by the applications
- Human resources and procedures for implementing the IT project

Various IT tools and methodologies are used to support the creation of an IT application architecture. The results obtained from Step 2 are routed to the strategic planning level; for example, to a steering committee. Based on the results of Step 2, the application portfolio (a portfolio is *a set of applications*) or a specific project may be changed. For example, the steering committee may scale down a specific project because it is too risky at that time. Once the architecture is compiled and the project gets final approval, a decision about *how* to acquire the specific IT application must be made.

IT at Work 13.1

High-Tech Census Project Fails—An Analysis

U.S. Secretary of Commerce Carlos M. Gutierrez issued the following official statement explaining (in an obscure way) why the Census Bureau was scrapping its $600 million project that was to collect data using 500,000 handheld devices. The bureau had contracted to use handheld devices from Harris Corp., but mismanagement, cost overruns, and poor planning helped derail the plan.

> *Over the last month or so, a clear sense has emerged: to have a fully successful 2010 Census, we must immediately revamp some programs, refocus priorities and get on top of the challenge.*
>
> According to a press release (*census.gov/*):
>
> *Multiple internal and external reviews have identified continuing Census challenges across a number of areas, including adequate planning over key systems requirements, key technology requirements, specification of operational control system characteristics and functions and regional center technology infrastructure . . . . Gutierrez said that the Census Bureau will need an additional $2.2 to $3.0 billion in funding over the next five years to meet the replan needs . . . . The life cycle cost for the Reengineered 2010 Census was estimated at $11.8 billion in the FY 2009 Budget Request, including $1.8 billion for the American Community Survey which replaced the long-form. The new estimated life cycle cost for the 2010 Census is $13.7 to $14.5 billion.*

In summary, the Census Bureau had planned to issue more than 500,000 handhelds to temporary employees to collect personal data on Americans who do not return census forms in the mail. The handhelds were being developed under a $600 million contract awarded to Harris Corp. in 2006. Stumbling over this multibillion-dollar plan for a high-tech census, the government reverted to counting the nation's 300 million people the old-fashioned way: with paper and pencil. Poor management—not poor technology—caused the government to spend an additional $3 billion for the next census.

Was the Failure a Surprise? Senator Susan Collins, ranking member of the Committee on Homeland Security and Governmental Affairs, wasn't surprised by the failure. "This committee is unfortunately no stranger to tales of federal projects and contracts that have gone awry, often at a heavy cost in taxpayer funds," she said. Collins listed the usual failure reasons:

- Poorly defined initial requirements
- Inability or unwillingness of management to control "requirements creep" and cost overruns

Something larger than poor project management was at work. It was the failure of top management in the bureau to assess and mitigate the risks inherent in such a major project. "It should be noted that the problems with this contract seemed apparent to everyone except the Census Bureau," said Senator Tom Coburn (D-Okla.).

Analysis of the Handheld Project Failure. The 2010 Census was to have been the first true high-tech count in the nation's history. The Census Bureau had awarded a contract to purchase 500,000 of the computers, plus the computer operating system, at a cost of more than $600 million. The contract ballooned to $1.3 billion, even though the bureau scaled back its purchase to only 151,000 handheld computers. The higher expenditure was due to cost overruns and new features ordered by the Census Bureau on the computers and the OS. Gutierrez blamed many of the problems on "a lack of effective communication with one of our key contractors."

Census officials were being blamed for doing a poor job of spelling out technical requirements to the contractor, Harris. In addition, the handhelds proved too complex for some temporary workers, who tried to use them in a test in North Carolina, and the devices were not initially programmed to transmit the large amounts of data necessary.

Harris spokesman Marc Raimondi said the cost of the contract increased as the project requirements increased. "The increased funding is required to cover additional sites, equipment, software and functions added by the bureau to the program."

Representative Alan Mollohan, chairman of the appropriations subcommittee, said the Census Bureau and Harris "contributed to today's crisis." The Census Bureau's failure to address problems with the computers early on "turned the crisis into the emergency that we now face."

Sources: Compiled from U.S. Census Bureau (2008), Hogue (2008), and Holmes (2008).

Discussion Questions: What went wrong? Make a list of things that went wrong and classify them as technology-related, management-related, and/or project-related (due to changes in the scope of the project). Consider the statement: "Hope is not a plan." Does that statement apply to this project failure? Explain why or why not.

Step 3: Selecting an acquisition option. IT applications can be:

- Built in-house. In-house development using the systems development life cycle (SDLC) approach is covered in Section 13.4.
- Custom-made by a vendor.
- Bought and customized, in-house or through a vendor. See Table 13.2 for a list of advantages and limitations of the *buy option.*

TABLE 13.2 Advantages and Limitations of the Buy Option

Advantages of the Buy Option	Disadvantages of the Buy Option
• Many different types of off-the-shelf software are available. • Much time can be saved by buying rather than building. • The company can know what it is getting before it invests in the software. • The company is not the first and only user. • Purchased software may preclude the need to hire personnel specifically dedicated to a project. • The vendor updates the software frequently. • The price is usually much lower for a buy option.	• Software may not exactly meet the company's needs. • Software may be difficult or impossible to modify, or it may require huge business process changes to implement. • The company will not have control over software improvements and new versions. (Usually it may only recommend.) • Purchased software can be difficult to integrate with existing systems. • Vendors may drop a product or go out of business

- Leased from an application service provider (ASP) or leased through a software-as-a-service (SaaS) arrangement, as you read in Chapter 12.
- Acquired via a partnership or alliance that will enable the company to use someone else's application.

Once an option is chosen, the system can be acquired. At the end of this step, an application is ready to be installed and deployed. No matter what option is chosen, you most likely will have to select one or more vendors and consulting companies.

Step 4: Testing, installing, integrating, and deploying IT applications. Once an acquisition option has been selected, the next step involves getting the application up and running on the selected hardware and network environment. One of the steps in installing an application is connecting it to back-end databases, to other applications, and often to partners' information systems. This step can be done in-house or outsourced. During this step, the modules that have been installed need to be tested. A series of tests are required:

- *Unit testing*: Testing the modules one at a time
- *Integration testing*: Testing the combination of modules interacting with other applications
- *Usability testing*: Testing the quality of the user's experience when interacting with the portal or Web site
- *Acceptance testing*: Determining whether the application meets the original business objectives and vision.

After the applications pass all of the tests, they can be rolled out to the end users. Here developers have to deal with issues such as conversion from the old to the new system, training, changes in priorities affecting acceptance of the application, and resistance to changing processes to maximize the benefit from the application.

Step 5: Operations, maintenance, and updating. It usually takes as much time, effort, and money to operate and maintain an application as it does to acquire and install it in the first place. For the maximizing of its continual usage, an application needs to be continually updated. Software maintenance can be a big problem due to rapid changes in the IT field. Operation and maintenance can be done in-house and/or outsourced.

Managing the IT Acquisition Process. The IT acquisition process most likely will be a complex project that must be managed properly. Except for small applications, an IT project team is usually created to manage the process, budget, costs, and vendors. Projects can be managed with *project management* software, such as Microsoft Project (*office.microsoft.com/project*). Three criteria that are used to evaluate the effectiveness of IT project management are performance, time, and cost. That is, was the IT project done right, on budget, and on time?

Standard project management techniques and tools are used by project managers to manage project resources to keep them on time, on budget, and within performance specifications. Finally, implementing an IT project may require restructuring one or more business processes.

IN-HOUSE DEVELOPMENT: INSOURCING

A third development strategy is to develop or build applications in-house. Although in-house development—*insourcing*—can be time-consuming and costly, it may lead to IT applications that better fit an enterprise's strategy and vision and differentiate it from competitors. The in-house development of IT applications, however, is a challenging task, as most applications are novel and may involve multiple organizations.

Options for In-House Development. Three major options exist for in-house development:

- **Build from scratch.** This option should be considered only for specialized IT applications for which components are not available. This option is expensive and slow, but it will provide the best fit to the organization's needs.
- **Build from components.** The required applications are often built from standard components (e.g., random number generators or Web servers such as Microsoft's IIS). Commercially packaged and homegrown components must integrate correctly and reliably or the system will fail. This is especially critical for real-time applications and for e-business systems. The scope of component integration and code reuse is broadening, too.
- **Integrating applications.** The application integration option is similar to the build-from-components option, but instead of components being used, entire applications are employed. This is an especially attractive option when IT applications from several business partners need to be integrated. Integration methods such as Web services or enterprise application integration (EAI) can be used.

Insourcing is a challenging task that requires specialized IT procedures and resources. For this reason, most organizations usually rely on packaged applications or outsource the development and maintenance of their IT applications.

Methods Used in In-House Development. Several methods can be used when developing IT applications in-house. Two major development methods are:

- **Systems development life cycle (SDLC).** Large IT projects, especially ones that involve infrastructure, are developed according to the SDLC methodology using several tools. Details about this approach are provided in Section 13.4.
- **Prototyping methodology.** With a prototyping methodology, an initial list of basic system requirements is defined and used to build a prototype. The prototype is then improved in several iterations, based on users' feedback. This approach can be very rapid. The prototype is then tested and improved, tested again, and developed further, based on the users' feedback. The prototyping approach, however, is not without drawbacks. There is a risk of getting into an endless loop of prototype revisions, as users may never be fully satisfied. Such a risk should be planned for because of the rapid changes in IT and business models.
- **Web 2.0 or Application 2.0 methodology.** This development approach involves quick, incremental updates with close user involvement. For new application developments, a beta (prototype) version is developed and then refined—also in very close collaboration with users.

End-User Development. **End-user development** (also known as **end-user-computing**) is the development and use of ISs by people outside the IS department. This includes users in all functional areas at all skill levels and organizational levels: managers, executives, staff, secretaries, and others. End-user development has risks and limitations. End users may not be skilled enough in computers, so quality and cost may be jeopardized unless proper controls are installed. Also, many end users do not take time to document their work and may neglect proper security measures.

Review Questions

1. What is the advantage of loosely coupled software design?
2. Explain the three-tier software architecture design.
3. Explain the functions of middleware.
4. What is IT architecture?
5. What testing needs to be done on an application?
6. List the major acquisition and development strategies.
7. Compare the buy option against the lease option.
8. List the in-house development approaches.
9. What are the risks and limitations of end-user development?

13.3 IT Project Management

Successful organizations perform projects that produce desired results in established timeframes with assigned resources. Projects are not limited to IT, but can apply to most all functions of the organization. The project management principles and practices discussed in this section apply to any type of project.

Projects are managed by managing the triple constraints, which are:

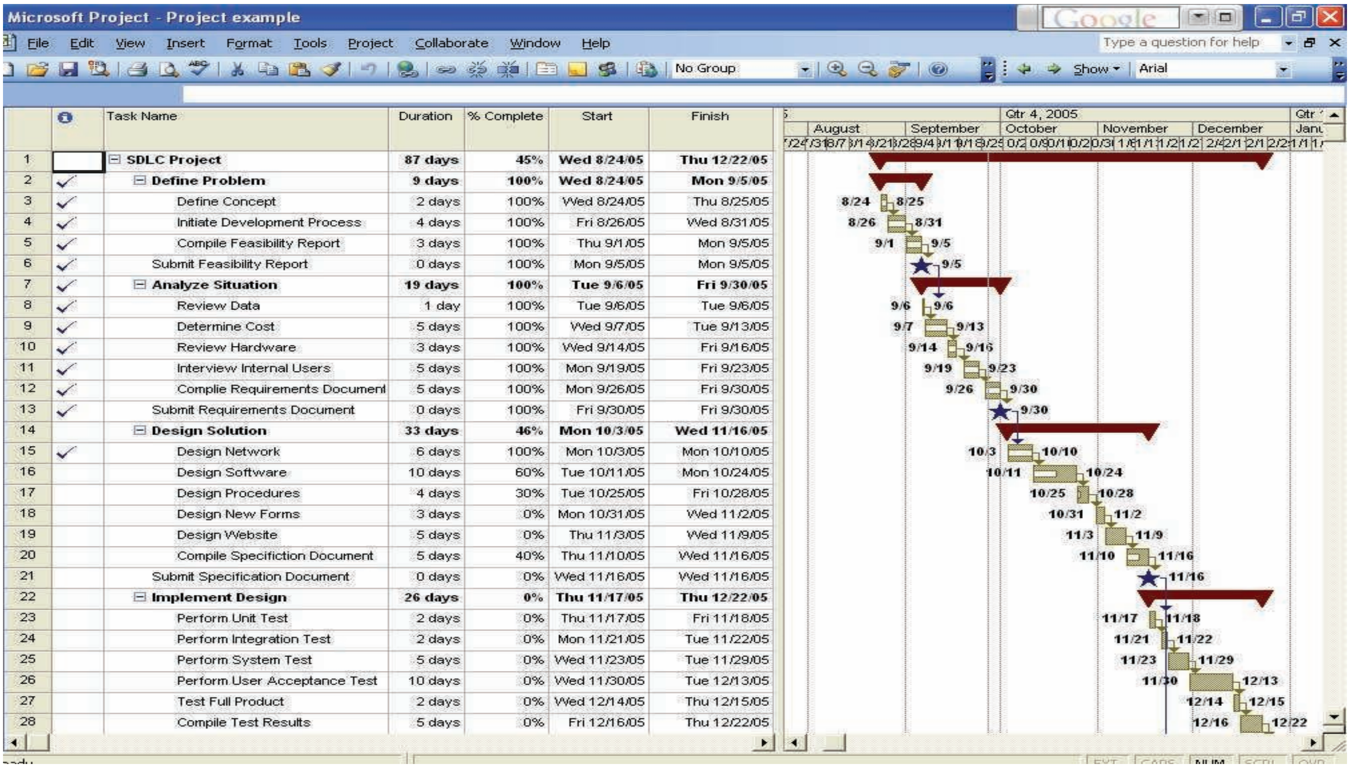
1. **Scope:** The project scope is the definition of what the project is supposed to accomplish—its outcomes or deliverables. Scope is measured in terms of the project size, goals, and requirements.
2. **Time:** A project is made of up *tasks.* Definitions of the tasks should start with active verbs, such as *purchase* servers, *apply for* permits, and *interview* vendors. Each task is assigned a duration, which is the difference between the task's start date and its end date. The project's time is determined by task durations and task dependencies. Some tasks are dependent on other tasks being completed before they can begin. For example, in construction, a hole must be dug before the pouring of concrete can start. Task durations and task dependencies determine the time required to complete the project.
3. **Budget:** Projects are approved subject to their costs.

These constraints are interrelated, so they must be managed together for the project to be completed on time, within budget, and to specification (spec).

After the project scope has been defined, it is used to estimate a realistic timeline and budget based on the availability of necessary resources. Resources include the people, equipment, and material needed to complete the project. The result is a project plan that is specified in a **Work Breakdown Structure (WBS).** Figure 13.5 shows a screen shot of Microsoft Project, with a WBS on the left side and a Gantt chart on the right side. A **Gantt chart** is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the Work Breakdown Structure of the project. Project resources must be managed according to the WBS.

Scope Creep. It is absolutely imperative that any change to the scope of the project explicitly include compensating changes in the budget, the deadline, and/or resources. **Scope creep,** which refers to the growth of the project after the scope has been defined, is a serious issue. Scope creep is the piling up of small changes that by themselves are manageable but in aggregate are significant. IT projects, particularly one as complex as implementing an ERP or CRM, can take a long time to complete.

During the project, it's almost guaranteed that requests will be made that change the scope. If the project scope is to build an accounting app for processing expense



		Task Name	Duration	% Complete	Start	Finish
1		⊟ **SDLC Project**	**87 days**	**45%**	**Wed 8/24/05**	**Thu 12/22/05**
2	✓	⊟ **Define Problem**	**9 days**	**100%**	**Wed 8/24/05**	**Mon 9/5/05**
3	✓	Define Concept	2 days	100%	Wed 8/24/05	Thu 8/25/05
4	✓	Initiate Development Process	4 days	100%	Fri 8/26/05	Wed 8/31/05
5	✓	Compile Feasibility Report	3 days	100%	Thu 9/1/05	Mon 9/5/05
6	✓	Submit Feasibility Report	0 days	100%	Mon 9/5/05	Mon 9/5/05
7	✓	⊟ **Analyze Situation**	**19 days**	**100%**	**Tue 9/6/05**	**Fri 9/30/05**
8	✓	Review Data	1 day	100%	Tue 9/6/05	Tue 9/6/05
9	✓	Determine Cost	5 days	100%	Wed 9/7/05	Tue 9/13/05
10	✓	Review Hardware	3 days	100%	Wed 9/14/05	Fri 9/16/05
11	✓	Interview Internal Users	5 days	100%	Mon 9/19/05	Fri 9/23/05
12	✓	Complie Requirements Document	5 days	100%	Mon 9/26/05	Fri 9/30/05
13	✓	Submit Requirements Document	0 days	100%	Fri 9/30/05	Fri 9/30/05
14		⊟ **Design Solution**	**33 days**	**46%**	**Mon 10/3/05**	**Wed 11/16/05**
15	✓	Design Network	6 days	100%	Mon 10/3/05	Mon 10/10/05
16		Design Software	10 days	60%	Tue 10/11/05	Mon 10/24/05
17		Design Procedures	4 days	30%	Tue 10/25/05	Fri 10/28/05
18		Design New Forms	3 days	0%	Mon 10/31/05	Wed 11/2/05
19		Design Website	5 days	0%	Thu 11/3/05	Wed 11/9/05
20		Compile Specifiction Document	5 days	40%	Thu 11/10/05	Wed 11/16/05
21		Submit Specification Document	0 days	0%	Wed 11/16/05	Wed 11/16/05
22		⊟ **Implement Design**	**26 days**	**0%**	**Thu 11/17/05**	**Thu 12/22/05**
23		Perform Unit Test	2 days	0%	Thu 11/17/05	Fri 11/18/05
24		Perform Integration Test	2 days	0%	Mon 11/21/05	Tue 11/22/05
25		Perform System Test	5 days	0%	Wed 11/23/05	Tue 11/29/05
26		Perform User Acceptance Test	10 days	0%	Wed 11/30/05	Tue 12/13/05
27		Test Full Product	2 days	0%	Wed 12/14/05	Thu 12/15/05
28		Compile Test Results	5 days	0%	Fri 12/16/05	Thu 12/22/05

Figure 13.5 Microsoft Project screenshot of WBS (left side) and Gantt chart (right side).

reports with a budget of $100,000 and a four-month duration, the project manager is expected to do that. However, if the scope is changed to also include processing of sales commissions, the project manager must obtain an appropriate change in budgeted resources and time. If the budget is not adjusted, the smart project manager will refuse to agree to the change in scope. Make sure any requested change, no matter how small, is accompanied by approval for a change in budget or schedule or both.

WHAT DO PROJECT MANAGERS DO?

Project management is the process of guiding a project from its beginning through its performance to its closure. Project management includes three basic operations:

1. **Planning:** Specifying the desired results, determining the schedules, and estimating the resources
2. **Organizing:** Defining people's roles and responsibilities
3. **Controlling:** Tracking the planned performance and budget against the actual performance; also, managing people's performances, addressing problems, putting out fires, and keeping priorities well known

Managing the Critical Path. Tasks must be completed in a specific order to get the job done. Certain tasks make up what is called the critical path, which is an important principle of project management. Project managers must manage the critical path. The **critical path** consists of activities or tasks that must start and finish on schedule or else the project completion will be delayed—unless action is taken to expedite one or more critical tasks. The critical path is the length of the project. Each task on the critical path is a **critical task.**

There are non-critical paths composed of tasks that are not critical, but since their status could easily change to critical, you need to monitor and manage the critical and non-critical paths.

The purpose of the **critical path method (CPM)** is to recognize which activities are on the critical path so that you know where to focus your efforts. You use critical tasks to identify or prioritize trade-offs.

Project Manager Success Skills. The success of a project manager depends on:

- **Communication:** Clear, open, and timely sharing of information with appropriate individuals and groups is necessary. Since people tend to want to admit bad news, extra effort is needed to ensure that news about anything that will delay or compromise the project is reported promptly. Without truthful and complete communication during the project, it will fail.
- **Information:** There should be no surprises. Accurate, timely, and complete data for the planning, performance monitoring, and final assessment is necessary.
- **Commitment:** Team members should personally promise to produce the agreed-upon results on time and within budget.

Review Questions

1. Define *triple constraint*.
2. What is the project scope?
3. What is scope creep? Why does it pose such a risk to the project and project manager?
4. What is the critical path?
5. What do project managers do?

13.4 Systems Development

The **systems development life cycle (SDLC)** is the traditional systems development method used by organizations for large IT projects such as IT infrastructure. The SDLC is a structured framework that consists of sequential processes by which information systems are developed. As shown in Figure 13.6, these processes are investigation, analysis, design, programming, testing, implementation, operation, and maintenance. The processes, in turn, consist of well-defined tasks. Large projects typically require all of the tasks, whereas smaller development projects may require only a subset of the tasks.

Within the SDLC, there is an iterative feature. *Iteration* is the revising of the results of any development process when new information makes this revision the smart thing to do. Iteration does not mean that developments should be subjected to infinite revisions, but it does mean that developers should adjust to new relevant information. Recall the scope creep that tends to happen to projects. IS design is highly susceptible

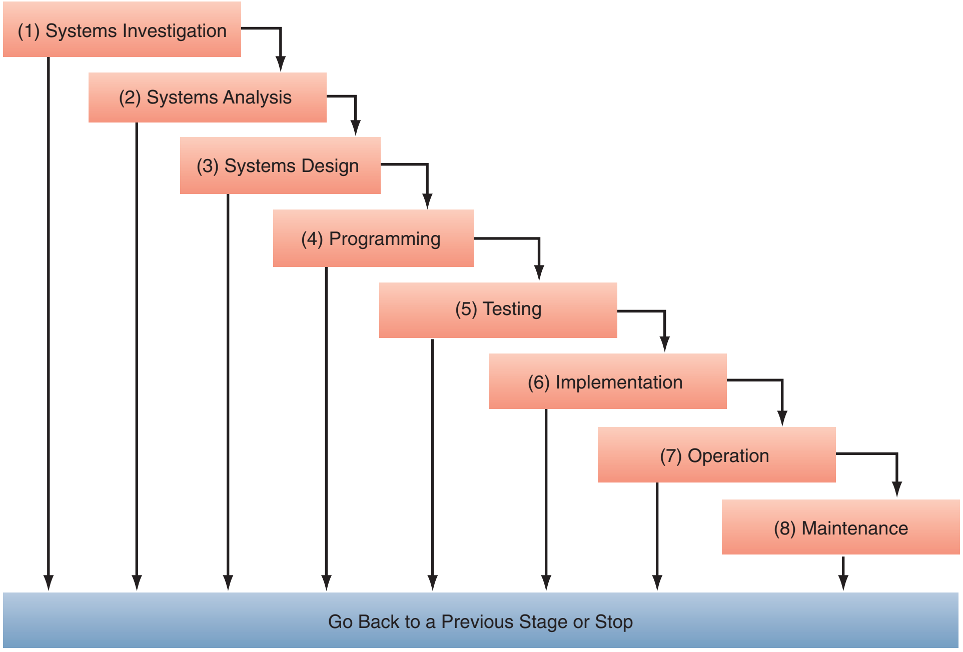


Figure 13.6 An eight-stage system development life cycle (SDLC).

to scope creep as users ask for additional features or try to keep up with the latest mobile technologies. It is especially important for social media, viral marketing, and e-commerce development because those systems must constantly evolve.

Systems development projects produce desired results through team efforts. Development teams typically include users, systems analysts, programmers, and technical specialists. *Users* are employees from all functional areas and levels of the organization who will interact with the system, either directly or indirectly. **Systems analysts** are information systems professionals who specialize in analyzing and designing information systems. Programmers are information systems professionals who modify existing computer programs or write new computer programs to satisfy user requirements. Technical specialists are experts on a certain type of technology, such as databases or telecommunications. All people who are affected by changes in information systems (e.g., users and managers) are known as systems stakeholders and are typically involved in varying degrees and at various times in the systems development.

SDLC STAGE 1: SYSTEMS INVESTIGATION

Systems development practitioners agree that the more time invested in understanding the business problem or opportunity, in understanding technical options for systems, and in understanding problems that are likely to occur during development, the greater the probability that the IS will be a success. For these reasons, systems investigation begins with the *business problem* or *opportunity*.

Problems and opportunities often require not only understanding them from the internal point of view, but also seeing them as organizational partners—suppliers or customers—would see them. Another useful perspective is that of competitors. How have they responded to similar situations, and what outcomes and additional opportunities have materialized? Creativity and out-of-the-box thinking can pay big dividends when isolated problems can be recognized as systemic failures whose causes cross organizational boundaries. Once these perspectives can be gained, those involved can also begin to better see the true scope of the project and propose possible solutions. Then an initial assessment of these proposed system solutions can begin.

Feasibility Studies. The next task in the systems investigation stage is the feasibility study. The feasibility study determines the probability of success of the proposed project and provides a rough assessment of the project's technical, economic, organizational, and behavioral feasibility. The feasibility study is critically important to the systems development process because, done properly, the study can prevent organizations from making costly mistakes, such as creating systems that will not work, that will not work efficiently, or that people cannot or will not use. The Census failure described in *IT at Work 13.1* is a good example. The various feasibility analyses also give the stakeholders an opportunity to decide what metrics to use to measure how a proposed system meets their various objectives.

- ***Technical feasibility.*** Technical feasibility determines whether the hardware, software, and communications components can be developed and/or acquired to solve the business problem. Technical feasibility also determines whether the organization's existing technology can be used to achieve the project's performance objectives.
- ***Economic feasibility.*** Economic feasibility determines whether the project is an acceptable financial risk and whether the organization can afford the expense and time needed to complete the project. Economic feasibility addresses two primary questions: Do the benefits outweigh the costs of the project? Can the project be completed as scheduled?

Three commonly used methods to determine economic feasibility are return on investment (ROI), net present value (NPV), and breakeven analysis. Return on investment is the ratio of the net income attributable to a project divided by the average assets invested in the project. The net present value is the net amount by which project benefits exceed project costs, after allowing for the cost of capital and the time value of money. Breakeven analysis determines the point at which the cumulative cash flow from a project equals the investment made in the project.

Determining economic feasibility in IT projects is rarely straightforward, but it often is essential. Part of the difficulty stems from the fact that benefits are often intangible. Another potential difficulty is that the proposed system or technology may be "cutting edge," and there may be no previous evidence of what sort of financial payback is to be expected.

- ***Organizational feasibility.*** Organizational feasibility has to do with an organization's ability to accept the proposed project. Sometimes, for example, organizations cannot accept a financially acceptable project due to legal or other constraints. In checking organizational feasibility, one should consider the organization's policies and politics, including impacts on power distribution, business relationships, and internal resources availability.
- ***Behavioral feasibility.*** Behavioral feasibility addresses the human issues of the project. All systems development projects introduce change into the organization, and people generally fear change. Overt resistance from employees may take the form of sabotaging the new system (e.g., entering data incorrectly) or deriding the new system to anyone who will listen. Covert resistance typically occurs when employees simply do their jobs using their old methods.

Behavioral feasibility is concerned with assessing the skills and the training needed to use the new IS. In some organizations, a proposed system may require mathematical or linguistic skills beyond what the workforce currently possesses. In others, the workforce may simply need to improve their skills. Behavioral feasibility is as much about "can they use it" as it is about "will they use it."

After the feasibility analysis, a "go/no-go" decision is reached. The functional area manager for whom the system is to be developed and the project manager sign off on the decision. If the decision is "no-go," the project is put on the shelf until conditions are more favorable or the project is discarded. If the decision is "go," then the systems development project proceeds and the systems analysis phase begins.

SDLC STAGE 2: SYSTEMS ANALYSIS

The systems analysis stage produces the following information: (1) strengths and weaknesses of the existing system, (2) functions that the new system must have to solve the business problem, and (3) user information requirements for the new system. Armed with this information, systems developers can proceed to the systems design stage.

There are two main approaches in systems analysis: the traditional (structured) approach and the object-oriented approach. The traditional approach emphasizes *how*, whereas the object-oriented approach emphasizes *what.*

SDLC STAGE 3: SYSTEM DESIGN

Systems analysis describes what a system must do to solve the business problem, and *systems design* describes *how* the system will accomplish this task. The deliverable of the systems design phase is the technical design that specifies the following:

- System outputs, inputs, and user interfaces
- Hardware, software, databases, telecommunications, personnel, and procedures
- How these components are integrated

This output represents the set of *system specifications*. Systems design encompasses two major aspects of the new system: **Logical system design** states what the system will do, using abstract specifications. **Physical system design** states how the system will perform its functions, with actual physical specifications. Logical design specifications include the design of outputs, inputs, processing, databases, telecommunications, controls, security, and IS jobs. Physical design specifications include the design of hardware, software, database, telecommunications, and procedures. For example, the logical telecommunications design may call for a wide area network connecting the company's plants. The physical telecommunications design will specify the types of communications hardware (computers and routers), software (network operating system), media (fiber optics and satellite), and bandwidth (e.g., 100 Mbps).

When both of these aspects of system specifications are approved by all participants, they are "frozen." That is, once the specifications are agreed upon, they should not be changed. However, users typically ask for added functionality in the system (called *scope creep*). This occurs for several reasons: First, as users more clearly understand how the system will work and what their information and processing needs are, they see additional functions that they would like the system to have. Also, as time passes after the design specifications are frozen, business conditions often change, and users ask for added functionality. Finally, because scope creep is expensive, project managers place controls on changes requested by users. These controls help to prevent *runaway projects*—systems development projects that are so far over budget and past deadline that they must be abandoned, typically with large monetary loss.

SDLC STAGE 4: PROGRAMMING

Systems developers utilize the design specifications to acquire the software needed for the system to meet its functional objectives and solve the business problem. Organizations may buy the software or construct it in-house.

Although many organizations tend to purchase packaged software, many other firms continue to develop custom software in-house. For example, Walmart and Eli Lilly build practically all of their software in-house. The chief benefit of custom development is systems that are better suited than packaged applications to an organization's new and existing business processes. For many organizations, custom software is more expensive than packaged applications. However, if a package does not closely fit the company's needs, the savings are often diluted when the information systems staff or consultants must extend the functionality of the purchased packages.

If the organization decides to construct the software in-house, then programming begins. **Programming** involves the translation of the design specifications into computer code. This process can be lengthy and time-consuming because writing computer code remains as much an art as a science. Large systems development projects can require hundreds of thousands of lines of computer code and hundreds of computer programmers. In such projects, programming teams are used. These teams often include functional area users to help the programmers focus on the business problem at hand.

In an attempt to add rigor (and some uniformity) to the programming process, programmers use structured programming techniques. These techniques improve the logical flow of the program by decomposing the computer code into *modules*, which are sections of code (subsets of the entire program). This modular structure allows for more efficient and effective testing because each module can be tested by itself. The structured programming techniques include the following restrictions:

- Each module has one, and only one, function.
- Each module has only one entrance and one exit. That is, the logic in the computer program enters a module in only one place and exits in only one place.
- GO TO statements are not allowed.

For example, a flowchart for a simple payroll application might look like the one shown in Figure 13.7. The figure shows the only three types of structures that are used in structured programming: sequence, decision, and loop. In the *sequence* structure, program statements are executed one after another until all of the statements in the sequence have been executed. The *decision* structure allows the logic flow to branch, depending on certain conditions being met. The *loop* structure enables the software to execute the same program, or parts of a program, until certain conditions are met (e.g., until the end of the file is reached or until all records have been processed).

As already noted, structured programming enforces some standards about how program code is written. This approach and some others were developed not only to improve programming, but also to standardize how a firm's various programmers do their work. This uniform approach helps ensure that all of the code developed by different programmers will work together. Even with these advances, however, programming can be difficult to manage.

SDLC STAGE 5: TESTING

Thorough and continuous testing occurs throughout the programming stage. Testing verifies that computer code works correctly under various conditions. Testing requires a lot of time, effort, and expense to do properly. However, the costs of improper

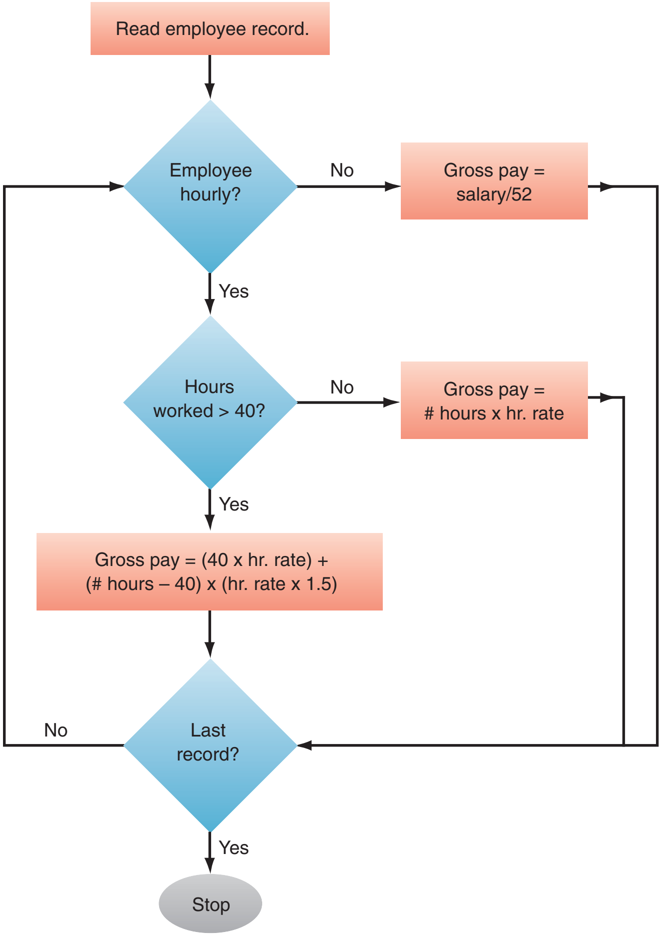


Figure 13.7 Flowchart of a payroll application.

testing, which could possibly lead to a system that does not meet its objectives, are enormous.

Testing is designed to detect errors (bugs) in the computer code. These errors are of two types: syntax errors and logic errors.

- *Syntax errors* (e.g., a misspelled word or a misplaced comma) are easier to find and will not permit the program to run.
- *Logic errors* permit the program to run but result in incorrect output. Logic errors are more difficult to detect because the cause is not obvious. The programmer must follow the flow of logic in the program to determine the source of the error in the output.

To have a systematic testing of the system, we must start with a comprehensive *test plan.* There are several types of testing: In *unit testing,* each module is tested alone in an attempt to discover any errors in its code. *String testing* puts together several modules to check the logical connection among them. The next level, *integration testing,* brings together various programs for testing purposes. *System testing* brings together *all* of the programs that comprise the system.

As software increases in complexity, the number of errors increases, making it almost impossible to find them all. This situation has led to *"good-enough" software,* software that developers release knowing that errors remain in the code but believing that the software will still meet its functional objectives. That is, they have found all the "show-stopper" bugs, errors that will cause the system to shut down or will cause catastrophic loss of data.

SDLC STAGE 6: IMPLEMENTATION

Implementation (or deployment) is the process of converting from the old system to the new system. Organizations use four major conversion strategies: parallel, direct, pilot, and phased.

In a **parallel conversion,** the old system and the new system operate simultaneously for a period of time. That is, both systems process the same data at the same time, and the outputs are compared. This type of conversion is the most expensive but also the least risky. Most large systems have a parallel conversion process to lessen the risk.

In a **direct conversion,** the old system is cut off and the new system is turned on at a certain point in time. This type of conversion is the least expensive but the most risky if the new system doesn't work as planned. Few systems are implemented using this type of conversion, due to the risk involved.

A **pilot conversion** introduces the new system in one part of the organization, such as in one plant or in one functional area. The new system runs for a period of time and is assessed. After the new system works properly, it is introduced in other parts of the organization.

A **phased conversion** introduces components of the new system, such as individual modules, in stages. Each module is assessed, and, when it works properly, other modules are introduced until the entire new system is operational.

Enterprise application integration (EAI) is often called the *middleware,* which you read in Section 13.2. Interfaces were developed to map the major packages to a single conceptual framework that guides what all these packages do and the kinds of information they normally need to share. This conceptual framework could be used to translate the data and processes from each vendor's package to a common language. It is the only way to implement collaborative supply chain sharing of information.

XML is the technology that is being used by many EAI vendors in their cross-enterprise applications development. It can be thought of as a way of providing variable format messages that can be shared between any two computer systems, as long as they both understand the format (tags) that is (are) being used.

SDLC STAGES 7 AND 8: OPERATION AND MAINTENANCE

After conversion, the new system will operate for a period of time, until it no longer meets its objectives. Once the new system's operations are stabilized, *audits* are performed during operation to assess the system's capabilities and determine if it is being used correctly.

Systems need several types of maintenance. The first type is *debugging* the program, a process that continues throughout the life of the system. The second type is *updating* the system to accommodate changes in business conditions. An example would be adjusting to new governmental regulations (such as tax rate changes). These corrections and upgrades usually do not add any new functionality; they are necessary simply in order for the system to continue meeting its objectives. The third type of maintenance *adds new functionality* to the system—adding new features to the existing system without disturbing its operation.

Review Questions

1. Define the eight stages of the SDLC.
2. What is the difference between logical and physical design?
3. Explain logic errors and syntax errors.
4. Explain the feasibility tests and their importance.
5. Discuss the four conversion methods.

Key Terms

BPM mashup *397*
business activity monitoring (BAM) *395*
business process *391*
business process management (BPM) *395*
business process reengineering (BPR) *395*
critical path *405*
critical path method (CPM) *406*
critical task *405*
end-user development (computing) *403*
Gantt chart *404*
IT architecture *401*
logical system design *408*
physical system design *408*
prototyping *403*
service-oriented architecture (SOA) *392*
systems development lifecycle (SDLC) *406*
Web services *397*
Work Breakdown Structure (WBS) *404*

Chapter Highlights and Insights

(Numbers refer to Learning Objectives)

1 New applications need to be connected to databases, other enterprise systems, and so on inside the organization. They may also be connected to partners' information systems. Web Services and service-oriented architecture (SOA) provide a way to reuse an organization's IT assets. Introducing new technology may require restructure or redesign of processes. Also, processes may need to be redesigned to fit standard software. Several methodologies exist for redesigning processes, notably BPR and BPM. IT can help in analyzing, combining, improving, and simplifying business processes.

2 The three-tier architecture simplifies application development and BPM. An organization's software architecture can also be designed for greater flexibility by using a tiered model.

3 Projects are managed by managing the triple constraints. The project scope is the definition of what the project is supposed to accomplish—its outcomes or deliverables. A project is made up of *tasks.*. Each task is assigned a duration, which is the difference between the task's start date and its end date. The project's time is determined by task durations and task dependencies. Some tasks are dependent on other tasks being completed before they can begin. For example, in construction, a hole must be dug before the pouring of concrete can start. Task durations and task dependencies determine the time required to complete the project. Projects have budgets that typically are exceeded due to scope creep.

4 Building ISs in-house can be done by using the SDLC or by using prototyping or other methodologies, and it can be done by outsourcers, the IS department employees, or end users.

Questions for Discussion

1. What is a business process and how does it differ from an information system?
2. Why is it important for all business managers to understand business processes?
3. Review the Microsoft International opening case. Why was it necessary to standardize common HR business processes?
4. Why did many early BPR efforts fail?
5. Explain the relationship between BPM and SOA.
6. Why is there confusion among IT practitioners as to what SOA is?
7. Why does BPM begin with understanding current processes?
8. Discuss the reasons why end-user-developed IT systems can be of poor quality. What can be done to improve the situation?
9. Explain the three-tier IT architecture.
10. Explain why IT is an important enabler of business process redesign.
11. What is the critical path?
12. Explain the triple constraint.
13. Explain the stages of the SDLC.
14. Why are feasibility tests done?

Exercises and Projects

1. Visit *www-01.IBM.com/software/info/bpm/* and download the IBM BPM eKit. After reviewing the eKit, explain how BPM and enterprise architecture improve business outcomes.
2. Research open-source BPM software vendors. Create a table that lists five of these vendors, the software applications they provide, and their features.
3. Refer to the prior exercise. Redo the research but on proprietary BPM vendors. In the table, list the pricing options if they are available.
4. Examine some business processes at your university or in your company. Identify one process that needs to be redesigned to eliminate waste or inefficiencies. Diagram the existing tasks/activities in the process, and then diagram an improved process. Use modeling tools in Microsoft Word or one of the free BPM or workflow software tools.
5. Explore project management software on vendors' Web sites. Select a single project management package, download the demo, and try it. Make a list of the important features of the package. Be sure to investigate its Web, repository, and collaboration features. Report your findings to the class.

Group Assignments and Projects

1. As a group, design an information system for a start-up business of your choice. Describe your chosen IT resource acquisition strategy, and justify your choices of hardware, software, telecommunications support, and other aspects of a proposed system.

2. Managing a project with Microsoft Project is often the approach to IT project management. But many users prefer to use Microsoft Excel instead. The main reasons are that MS-Project is too expensive, wastes too much time to set up and keep updated, and is tough to use. The debate between Excel and Project has valid arguments for either approach. Each of two groups takes one software tool—Excel or Project—and debates the advantages of their tool.

Internet Exercises

1. Visit Gartner at *gartner.com/technology/research/content/business_process_improvement.jsp*. Listen to the podcast "Tying BPM to Other IT Disciplines." The podcast is 11.5 minutes. Why is it important to link BPM to other IT disciplines?
2. Visit *ehow.com/how_4460942_use-excel-project-manage-ment.html* and read the four steps on "How to Use Excel for Project Management." You can also go to *ehow.com* and then search for this title. Then visit *ehow.com/video_2324033_plan-large-project.html* and read "How to Plan a Large Project" and watch the video. *Review* the capabilities of MS-Project for managing large projects. Prepare a report comparing the advantages and disadvantages of using Excel and MS-Project to manage a large IT project. In your comparison, assume that the IT project management team will consist of ten users, each user has Excel 2007 or 2010 and Internet access, and no one has MS-Project.
3. Search the Internet to find recent material on the role BPM software plays in support of BPM. Select two vendors' software products and download/view their demos. In your opinion, how useful were the software products? For example, would they be helpful for both simple processes and complex processes? What skill level was needed to use the tools? Were they easy to learn *how to use*?

BUSINESS CASE

Pep Boys Accelerates Its Planning Process

Pep Boys (*pepboys.com/*) is a $2 billion publicly held company and the only aftermarket retail and service chain in the United States capable of serving all four segments of the automotive aftermarket: the do-it-yourself, do-it-for-me, buy-for-resale, and replacement tires. The company operates 582 stores in 35 states and Puerto Rico, is headquartered in Philadelphia, and has approximately 18,000 employees.

Budget Process Supported by Adaptive Planning Software

Pep Boys used Adaptive Planning (*adaptiveplanning.com/)* to prepare the budgets for its 580-plus automotive-service stores for its 2009 and 2010 fiscal years. The company previously had its budget information on Excel spreadsheets, which were sent to the store owners for changes, then to area directors for review, then back to headquarters for more changes and, ultimately, a consolidation.

"It was an administrative nightmare to create all those files, get them out there, follow up on their status, and collect them," according to Phil McAllister, director of budgeting and internal reporting.

With all users now accessing a single system running on top of a relational database and thus able to view updated information in real time, Pep Boys is saving 600 person-hours of effort per budgeting cycle compared with its former cumbersome procedure.

Selection Criteria

Pep Boys selected Adaptive Planning because it didn't need the largest vendors' more sophisticated software for purposes of its individual store budgets. McAllister notes that Pep Boys hasn't even come close to using all of the software's features.

A shorter implementation timeframe was also appealing, since the decision to seek a more collaborative budgeting process was made too late in 2008 to accommodate a drawn-out installation of an on-premises system. It took about six weeks for Adaptive Planning to create templates incorporating all the store-budget information and get the service up and running.

Adaptive Planning service's list price is $600 to $800 per user per year. Pep Boys has about 650 users, including the stores, area directors, division vice presidents, and administrators.

Sources: Compiled from *adaptiveplanning.com/index.php/* (2010) and *pepboys.com/about_pep_boys/* (2010),

Questions

1. Diagram the process by which Pep Boys had collected data during its budget planning process.
2. View Adaptive Planning's online demo at *adaptiveplanning.com/demo_recorded.php/*. This demo covers expense planning, personnel planning, sales planning, reporting, and administration. How does this software improve the various planning processes?
3. Evaluate Pep Boys' software vendor selection process. The outcome was successful, but given the selection process, what risks did the company take?
4. In your opinion, why did the Adaptive Planning company provide Pep Boys with excellent service and support?

PUBLIC SECTOR CASE

Financial Industry Regulatory Authority (FINRA) SOA Project

The Financial Industry Regulatory Authority (FINRA; *finra.org/*) is the largest independent regulator for all securities firms doing business in the United States. FINRA oversees nearly 4,700 brokerage firms, about 167,000 branch offices, and approximately 635,000 registered securities representatives. FINRA was created in July 2007 through the consolidation of NASD (National Association of Securities Dealers) and the regulatory and enforcement functions of the New York Stock Exchange (NYSE). FINRA protects investors and market integrity through effective and efficient regulation and complementary compliance and technology-based services (*finra.org/AboutFINRA/*).

FINRA SOA Project

The FINRA SOA project consolidated the NYSE member regulation systems with the NASD member regulation information systems. The primary challenges:

- The two organizations' application portfolios, which support the member regulation business, had to be consolidated. Each application portfolio was sizable and heterogeneous. At the onset, FINRA had 160 applications and NYSE member regulation had 86 applications.
- The two sets of legacy business processes had to be reconciled into a final-state business process.
- Final-state business processes had to seamlessly integrate new systems and existing systems from both legacy organizations. The existing systems required enhancements.
- Business teams were distributed across the United States in district office locations. The development team was located in New York City and the Washington, D.C., area.

The three key objectives were:

- The final-state business processes of the merged company required seamless operation.
- The team needed to ensure a continuity of business operations while transitioning in phases to the new final-state business processes.
- Performance and reliability of the systems was a key requirement in maintaining core mission success.

SOA selection criteria were:

- The size and complexity of the project required multiple teams in different locations working effectively in parallel to meet the aggressive schedule.
- An SOA approach reduced risks presented by the large team size.
- The end-state systems had to be flexible and provide the ability to quickly deploy changed and new business processes without breaking the architecture.
- It was anticipated that the approach would deliver significant savings in both cost and time when compared to competing approaches.

ROI of the SOA Approach

The member regulation function of FINRA (the new, merged regulator) benefited greatly from the new system. Broker regulation tasks were simplified and accelerated, and delivered cost savings for the business. The key business values achieved are:

1. Time-to-market: Project delivery was greatly accelerated by allowing development teams to conduct parallel development of 10 major services with minimal interaction and dependencies. The service-oriented approach and detailed overall vision allowed each team to rapidly deliver individual services that were seamlessly integrated and tested by the system team.
2. Reduced risk: The SOA approach mitigated many of the risks associated with large development teams (100+ staff) by facilitating parallel development while minimizing team interdependencies and setting clear team responsibilities. The key to reducing risk is the early definition of business service interfaces and responsibilities.
3. Cost savings: The modular SOA architecture of the new system consolidated business functions into a common set of business services that are leveraged across many business processes, resulting in cost savings for construction, deployment, and maintenance of the system.
4. Improved agility: The business-centric service design and modularity of the SOA approach provides flexible deployment to support current business processes and to rapidly adapt to support future business process. Current business-centric services include data sourcing, analytic surveillance, and case management.
5. Resilience: Fault-tolerant business continuity is achieved using guaranteed message delivery, as individual business services are moved off-line for maintenance and restored.
6. Process optimization: Technology duplication is eliminated through the consolidation of functionality into discrete standardized business services. This also provides a uniform approach and consistent results across all the business processes.

SOA Lessons

1. The single most important lesson from this SOA project is that extremely large, time-critical applications can utilize an SOA approach to segregate and compartmentalize common services and allow for massively parallel work by independent teams. This approach not only increases organizational productivity but also mitigates some of the risk presented by a large project.
2. The SOA approach gave the teams a measure of insulation, helping ensure that decisions on one component did not negatively impact other components or the project.

The decoupling allowed teams to deliver well-defined components on the aggressive timeline required for project success.

3. Understanding the underlying business problems and processes is crucial to creating well-defined services that are reusable and exhibit the correct level of granularity. The payoff for this is a flexible business process that can change and grow without changing the architecture.
4. A concise architectural vision shared between system architects and application architects is key in large projects. Effective governance, along with well-defined services with clear functions and interfaces, is essential. Since the interfaces will change over time, it is important to develop a plan to handle this early.

Sources: Compiled from FINRA (2010), SOA Consortium (2010), and *CIO Magazine* (2009).

Questions

1. How tolerant of business disruption or errors do you think FINRA would be? Explain your answer.
2. How many legacy systems needed to be integrated after the merger? Why would those systems be considered valuable assets?
3. Why did FINRA select SOA?
4. Based on this case, how would you explain SOA to someone who has never heard of it?
5. Why was this implementation a success? Do you think that cost was one of the most important concerns? Why or why not? Assuming that it was not *cost*, what do you think was the most important criteria during the implementation?

MODELING USING ARIS EXPRESS AND BLUEPRINT

Modeling a Business Process and Brainstorming a Business Strategy

ARIS Express is a BPM modeling tool based on industry standards. A free downloadable feature-rich version is available from *http://www.ariscommunity.com/aris-express/how-to-start*. Support features on the Web site are installation instructions, quick reference, and video tutorials.

1. To get started with ARIS, download and install ARIS Express.
2. View the video tutorial "How to Model Business Processes" to learn how to model process steps in ARIS Express and understand the meaning of symbols used in the "business process" model type. View other tutorials, as needed.
3. Create a new model type. Select business process as the model type.
4. Design and develop a model of a business process. Review your model for any missed steps or other omissions. Edit as needed.
5. Download the BPM Blueprint 30-day trial and the demo from *http://www-01.ibm.com/software/integration/bpm-blueprint/*. Model the process you'd completed in #4 with this software tool.
6. Which BPM modeling tool did you prefer? Why?
7. View the ARIS Express video tutorial "How to Model a Whiteboard" to learn how to structure ideas and tasks with a Whiteboard model. Then use the Whiteboard to model a brainstorming session related to a business plan. For example, you could brainstorm ideas about how to manage a new project to use of social media and 2D tags to market a new product or service. To what extent did the Whiteboard tool make planning the project easier? Explain.

Resources on the Book's Web Site

More resources and study tools are located on the Student Web site and on WileyPLUS. You'll find additional chapter materials and useful Web links. In addition, self-quizzes that provide individualized feedback are available for each chapter.

Case studies for Chapter 13 are available at wiley.com/college/turban:

13.1 NCBJ Achieves a 500 Percent ROI by Rebuilding Its IT Infrastructure

13.2 Con-way, Inc. Implements Innovative Technology and Wins Nationwide Recognition

13.3 Flickr's Application Development 2.0 Model

References

ARIS Express, *ariscommunity.com/aris-express/how-to-start/*

CIO Magazine, "SOA Consortium and CIO Magazine Announce Winners of SOA Case Study Competition*,"* 2009. *soa-consortium.org/*

Financial Industry Regulatory Authority (FINRA), 2010. *finra.org/About FINRA*

Hogue, F., "Handling the Census Handheld Debacle," *Baseline.com,* April 15, 2008.

Holmes, A., "Census Program to Use Handheld Computers Said to Be in 'Serious Trouble,'" *GovernmentExecutive.com,* January 2, 2008. *govexec.com/dailyfed/0108/010208h1.htm/*

Microsoft.com, 2010.

Oracle BAM, 2010*. oracle.com/appserver/business-activity-monitoring.html*

SOA Consortium, 2010. *blog.soa-consortium.org/*

U.S Census Bureau, Press Release 2008. *census.gov/*

Visio 2010, microsoft.com