CHAPTER SEVEN OUTLINE

STUDENT LEARNING OUTCOMES

1. Describe how a service-oriented architecture can be used as a philosophical approach to help the organization of the future meet all its IT-related needs.
2. Define and describe the various hardware and software infrastructure considerations in an organization.
3. Compare and contrast commonly used metrics for assessing the success of IT systems and IT-related initiatives.
4. Describe business continuity planning (BCP) and its phases.

PERSPECTIVES

WEB SUPPORT

www.mhhe.com/haag

- Best in computer statistics and resources
- Meta data
- Finding hosting services
- Searching for shareware and freeware

SUPPORTING MODULES

XLM/E Network Basics
Extended Learning Module E provides an introduction to the vast, exciting, and dynamic field of information technology networks. The module includes discussions of what is needed to set up a small network at home, the components used to build large business networks, Internet connection possibilities, types of communications media, and network security.

XLM/G Object-Oriented Technologies
Extended Learning Module G provides an introduction to the world of object-oriented technologies and concepts. Specifically, you will learn about the five primary object-oriented concepts, how classes and objects are related, and the three fundamental principles of object-oriented technologies.

CHAPTER SEVEN

Enterprise Infrastructure, Metrics, and Business Continuity Planning

Building and Sustaining the Dynamic Enterprise

OPENING CASE STUDY: NEITHER RAIN NOR SNOW NOR DARK OF NIGHT . . . IT'S NOT THE POST OFFICE—IT'S THE IRS

In June 2006, a three-foot high wall of water rushed down Constitution Avenue and crashed into the Internal Revenue Service (IRS) building. A moat approximately six feet deep formed around the building. The basement windows gave way to the force and weight of the water, which poured into the building with such force that furniture and equipment were thrown through doors and windows. With more than a foot of rain falling in a short period of time, the IRS found its building useless and 2,400 employees without an office.

It almost sounds like a Hollywood disaster action movie but it's real life and it has happened to more than just the IRS. Almost daily, a business somewhere suffers from some sort of disaster. It may be fire or flood, or it may be the loss of electricity for an extended period of time.

Whatever the case, all organizations need *business continuity plans* in place they can execute to continue doing business. Especially for their information technology and related assets, backup plans for doing business are essential. Most businesses can survive only a few days without IT systems, application software, and an underlying and supporting IT infrastructure.

The IRS, fortunately, did have a business continuity plan. Its main data centers are located outside the District of Columbia and were therefore unaffected by the water. Information is a key resource in any business, and the IRS was fortunate to have its people and technology working in one location and its information secured in another.

Several years ago, the IRS created a remote network access infrastructure, called Enterprise Remote Access Project Virtual Private Network and designed by AT&T, which allows employees to work from home and other locations while still having access to IRS data centers. Part of the IRS's business continuity plan calls for even more employees working via telecommuting. And that's what many employees did. They worked from home and they even came in to other federal buildings and used temporary workstations set up in every inch of available space, even conference rooms.

The lesson all businesses can learn from the IRS flood disaster is that it is vitally important to plan for disasters and to have a business continuity plan in place that can be executed to ensure that downtime is as minimal as possible.

Business continuity planning is just as important as the technological infrastructure it seeks to support in the event of a disaster. In this chapter, we'll explore IT infrastructures and associated considerations as well as how to build a solid business continuity plan.[1]

Questions

1. What sort of "personal continuity plan" do you have for your car, apartment, and other important parts of your life?
2. Are brick-and-mortar businesses or click-and-order businesses more susceptible to disasters and other interruptions? Justify your answer.
3. When was the last time you were dealing with a company and someone said, "I'm sorry but our computer systems are down right now, so I can't help you"? How did that make you feel? How did you react?

LEARNING OUTCOME 1

Introduction

In Chapter 6, we introduced you to the concept of a service-oriented architecture within the context of systems development. A ***service-oriented architecture (SOA or SoA)*** is a software architecture perspective that focuses on the development, use, and reuse of small self-contained blocks of code (called *services*) to meet all the application software needs of an organization. Thus new applications build on past solutions organically using established building blocks. If SoA is adopted, the organization is saying that instead of bringing in brand new systems all its software will be developed from reusable units of code. SoA is a high-level, holistic organizational approach to how an organization views and acts on all its software needs.

By way of illustration, let's move away from software development—and IT in general—for just a moment and consider the concept of your organization adopting a service-oriented architecture perspective for *everything* it does, such that those self-contained building-blocks of code (services) apply not just to software but also to people, processes, departments and units, operations, and best practices. What would that look like? Your organization with a service-oriented architecture philosophy would:

- Be a lean, agile organization that takes advantage of every resource in the most efficient and effective way.
- React quickly in a proactive way to perceived changes in the market, competition, and customer demographics, wants, and desires.
- Respond quickly to and adapt to new advances in technology.
- Transform its processes, structure, and HR initiatives to match a changing and dynamic workforce.

In short, SoA would enable your organization to become the organization of the future . . . bound by very few structural constraints, able to change on a moment's notice, always looking for and capitalizing on the next great competitive advantage.

Of course this is a book about how organizations use technology, so exploring a service-oriented architecture approach to things like HR or changing customer demographics is beyond our scope. So, let's refocus on IT referring to Figure 7.1, which outlines how an SoA perspective enables your organization to respond more adeptly to customers, end users, software development, information needs, and hardware requirements.

CUSTOMERS

An IT-enabled SoA philosophy allows your organization to provide customers with multichannel service delivery options and customizable products and services. Customers should be able to *plug-and-play* into any communications channel with your organization, such as fax, the Web, face-to-face contact, phone call, and so on. Regardless of the communications channel, the experience should be the same—a consistent and high-quality interaction with your organization.

Customers should also be able to interact with IT systems that allow them to customize and personalize products and services. By simply "plugging in" their desires and wants, your organization should respond with unique and individually tailored offerings that satisfy and delight the customer. We spoke about the concept of mass customization (*slivercasting* and *exclusivity*) in Chapter 5.

THE FOCUS	NOTES
CUSTOMERS	• Multi-channel service delivery • Consistent, high-quality interactions regardless of the venue • Customizable product and service capabilities
END USERS	• Fully integrated ERP system • Interoperability among vendors • Interoperability of modules by the same vendor • Mobile computing (access to information and software regardless of location and device)
SOFTWARE DEVELOPMENT	• SoA as a framework • RAD, XP, and agile as development methodologies • Exciting new deployments like Web 2.0
INFORMATION NEEDS	• End users with access to all types of information • Integrated information, business intelligence, and knowledge • Data warehouses • Standard information formats • Integrity controls • No duplicate information
HARDWARE REQUIREMENTS	• Integration of different technologies and technology platforms • Large storage capacity • Your focus on logical, not physical • Safe and secure telecommunications platform

Figure 7.1
A Service-Oriented Architecture (SoA) Philosophy

END USERS

An SoA philosophy requires that your organization view its end users of IT (i.e., employees within the organization) just as it does external customers. This is mainly achieved through a fully integrated enterprise resource planning (ERP) system that meets every application software and information need of each and every employee. The ERP system should support transparent interoperability across multiple vendors and within ERP modules provided by the same vendor. Because ERP systems are so vitally important to

today's integrated and agile organization, we'll further discuss them in this chapter in an upcoming section.

End users should as well be able to take advantage of multi-channel service delivery. In this case, end users (employees) should be able to access computing and information resources regardless of where they are (the notion of *mobile computing* from Chapter 5). And regardless of the IT device in hand (laptop, desktop, PDA, Web-enabled cell phone), employees should enjoy access to a full range of application software services and information.

SOFTWARE DEVELOPMENT

Organizations today can choose among numerous software development methodologies that focus on the production and reuse of blocks of code to speed the process of software development—rapid application development (RAD), extreme programming (XP), and agile methodology are among them.

If you delve into the nitty-gritty of software development, you'll find infrastructure platforms that support a service-oriented architecture approach. Some of these include Ajax (**A**synchronous **J**avaScript **a**nd **X**ML), SOAP (**S**ervice **O**riented **A**rchitecture **P**rotocol), WSDL (**W**eb **S**ervices **D**escription **L**anguage), UDDI (**U**niversal **D**escription, **D**iscovery, and **I**ntegration), and CORBA (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture). Many of these platforms support the Web 2.0 applications—such as wikis, blogs, and mashups—that we discussed in *Extended Learning Module B* and Chapter 4.

INFORMATION NEEDS

A service-oriented architecture philosophy leverages the most vitally important organizational resource—information. End users need access to all types of information on a moment's notice, regardless of where that information is located (or where the end user is located). People throughout your organization need access to information, business intelligence, and knowledge that supports their decision-making efforts. Recall from Chapter 3 that data warehouses are built by combining information from multiple sources. These data warehouses are of paramount importance to good decision making.

For those reasons and many more, an SoA approach to information requires that:

- Information be in a standard format no matter where it exists.
- Strict and rigorous integrity control mechanisms are in place to ensure the completeness, accuracy, and validity of the information.
- No duplicate information exists in disparate silos anywhere in your organization.
- Any kind of information from any source (even external) can be quickly and easily coupled with other information.

HARDWARE REQUIREMENTS

Finally, a service-oriented architecture philosophy must pervade all choices in the realm of hardware. Organizations should be free to choose different technologies and different technology platforms and integrate them seamlessly (i.e., plug-and-play). Powerful storage area networks should have the capacity to store all your information needs. And you should not have to care where within these networks information is stored; your access to the information should be simple and easy.

Your telecommunications platform should be safe and secure and, at the same time, enable you to access a network either wired or wirelessly using the same steps or

procedures. You should never have to look at the back of your computer to determine if you need to log on using a wired protocol or a wireless protocol.

That is what a service-oriented architecture philosophy is all about. Of course, the question now becomes how you implement such a philosophy. That's what the remainder of this chapter is about, specifically:

1. Hardware and software infrastructure
2. Metrics for determining success
3. Measures to ensure consistent, uninterrupted success

Hardware and Software Infrastructure

LEARNING OUTCOME 2

Generally, ***infrastructure*** is a relative term meaning "the structure beneath a structure." This definition implies different layers of structure, which provide support or services. An IT infrastructure is the implementation of an architecture—in our discussion here a service-oriented architecture. In a city, the infrastructure includes its streets and street lighting, hospitals, schools, utility lines running above and below the ground, and so on. We all depend on this public infrastructure to make lives of communities, cities, and people safe and prosperous. In a corporation, the IT infrastructure includes the hardware, software (such as ERP software), and information that (1) ensure the components work together and (2) enable people, business processes, and customers to interact and perform their tasks (see Figure 7.2).

ENTERPRISE RESOURCE PLANNING (ERP) REVISITED

ERP systems are big business. At the top of the IT spending list is the ERP market. For instance, the U.S. federal government will spend $7.7 billion on ERP products and services in fiscal year 2009, up 37 percent from 2004 spending of $5.6 billion. More than 60 percent of the Fortune 1000 companies have installed or are in the process of implementing ERP systems to support their business activities. These packages implemented by the Fortune 1000 companies run well over the IT budgets for most small-to-medium-size enterprises, and ERP vendors are targeting this untapped market with scaled-back

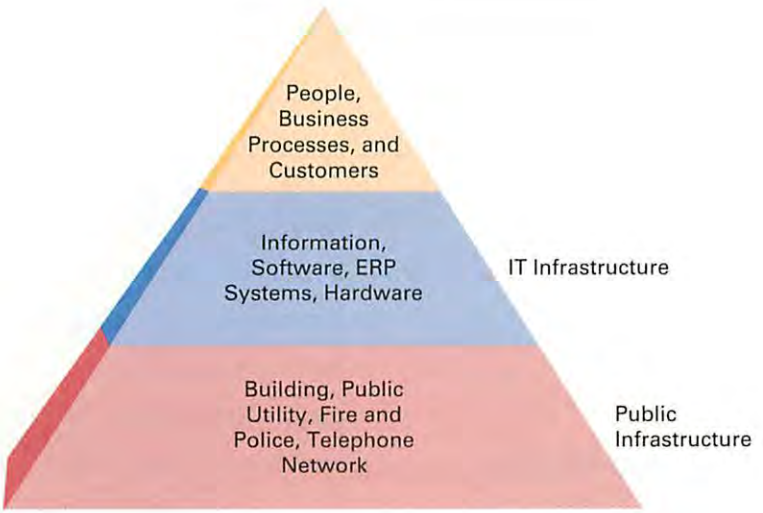


Figure 7.2
Infrastructure

systems suitable for smaller firms by offering simple, cheaper, preconfigured solutions easy to install within budget and time constraints. For instance, Microsoft now offers an ERP solution (called *Microsoft Great Plains*) geared toward the small-to-medium-size company.[2,3]

The dominating ERP software suppliers are SAP, Oracle/PeopleSoft, SSA Global, and Microsoft (see Figure 2.10 in Chapter 2 on page 81). Together they control more than 70 percent of the multibillion-dollar global market. Each vendor, for historical reasons, has a specialty in one particular module area such as SAP in logistics, Oracle/PeopleSoft in financials, SSA Global in manufacturing, and Microsoft in retail management.

There are also about 50 established and a few more newly emerging smaller and midsize ERP vendors including third-party developers competing for the very lucrative ERP market. Figure 7.3 displays a list of some other top ERP companies with links to their Web sites. There is stiff competition and overlapping products difficult to differentiate. The vendors are continuously updating their products and adding new technology-based features. Long-term vision, commitment to service and support, specialty features, experience, and financial strength for research and development are considered the major vendor qualities for product selection and implementation.

The ERP market has been growing at a rate of more than 30 percent per year and most forecasts predict more of the same. The growth of the ERP market has been boosted by both business and technical factors. With respect to business, the most cited reason is globalization, which has fostered mergers and stimulated the creation of big corporations with high information requirements that the former individual information systems were not able to fulfill. Another factor is general market maturity in developed countries, which has fostered competition among all companies and increased the power of consumers, thus forcing enterprises to upgrade the efficiency of their business processes. Finally, advances in information and communication technologies have made the development of ERP systems possible by allowing the database centralization to integrate with a distributed ERP environment.

A technical reason for ERP growth has been the introduction of the European Union (EU) currency, the euro, since most of the information systems in the EU zone were not able to handle multiple currencies.

THE EVOLUTION OF ERP SYSTEMS ERP systems replace "islands of information and processes" with a single, packaged software solution that integrates all the traditional enterprise management functions such as financials, human resources, and manufacturing and logistics. Knowing the history and evolution of ERP is essential to understanding its current application and its future developments. To help give you a better perspective, let's review the evolution of ERP systems (also see Figure 7.4).

Figure 7.3

Additional ERP Vendors and Their Web Addresses

Invensys www.invensys.com	Cincom www.cincom.com
Visibility www.visibility.com	Verticent www.verticent.com
i2 www.i2.com	Extensity www.extensity.com
QAD www.qad.com	Epicor www.epicor.com
IFS www.ifsab.com	Ceecom www.ceecom.com
Exact www.exactamerica.com	Ramco www.ramco.com

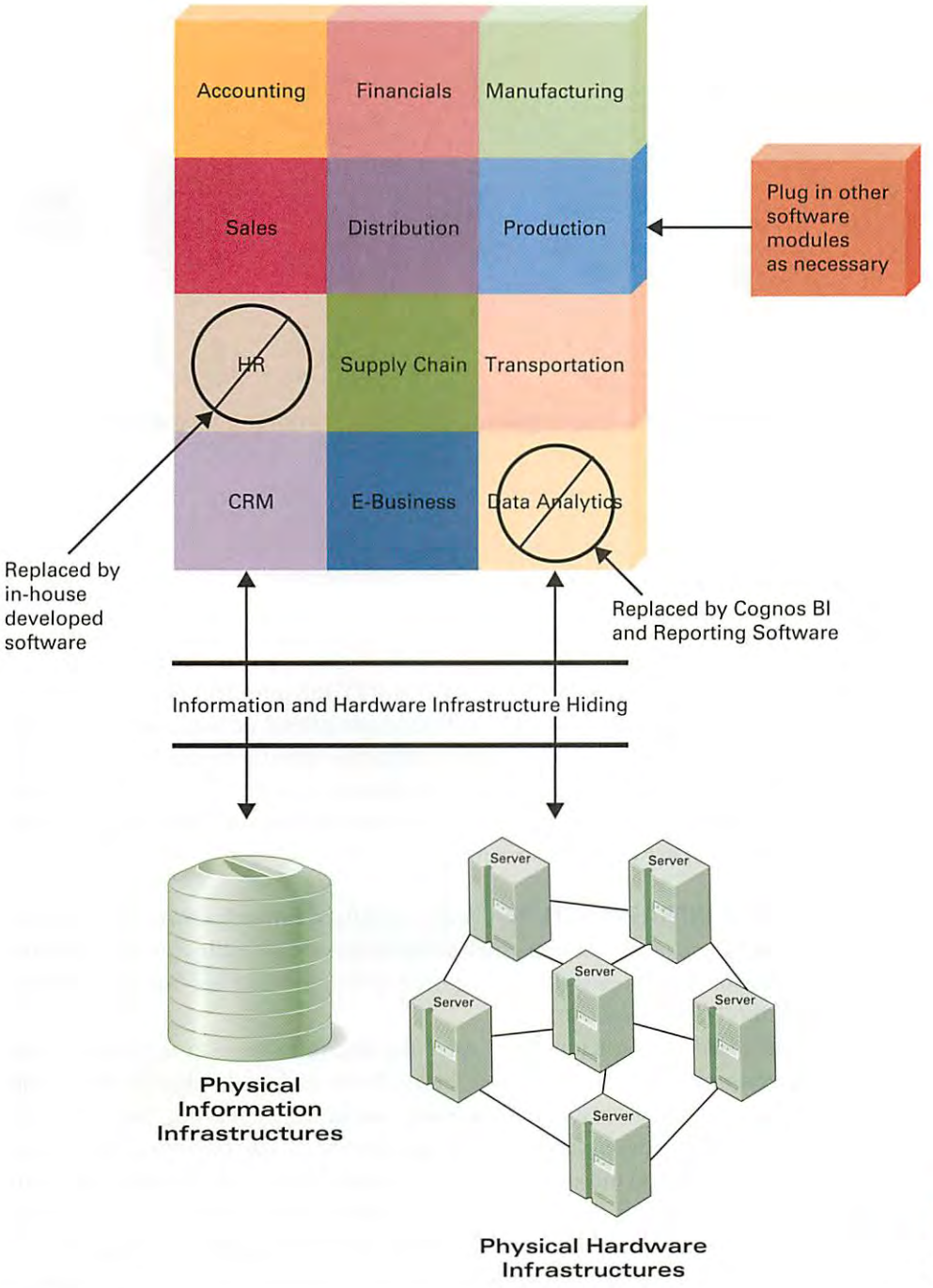


Figure 7.5
An ERP Infrastructure

Advantages

- Reliable information access
- Avoid data and operations redundancy
- Delivery and cycle time reduction
- Cost reduction
- Easy adaptability
- Improved scalability
- Global outreach
- E-business support

Disadvantages

- Time-consuming
- Expensive
- Lack of conformity of modules
- Vendor dependence
- Too many features, too much complexity
- Questionable scalability and global outreach
- Not enough extended ERP capability

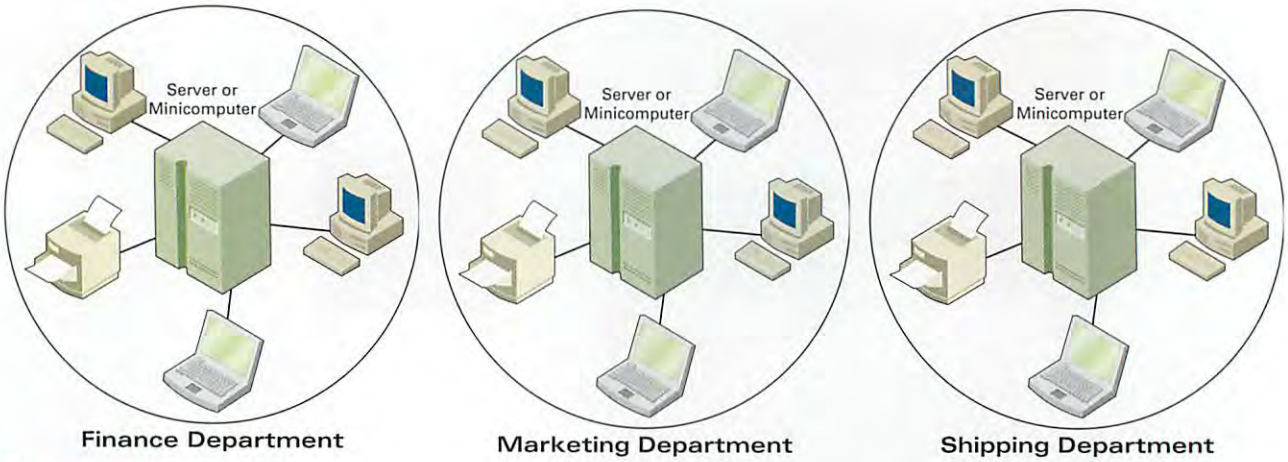


Figure 7.6

Decentralized Infrastructure

SUPPORTING NETWORK INFRASTRUCTURES

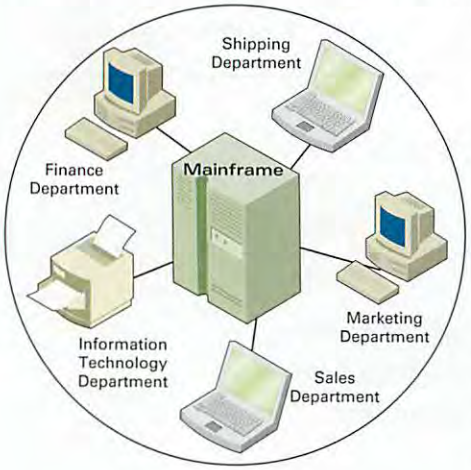
The fundamental underlying infrastructure for any IT environment is a network, two or more computers sharing information, software, peripheral devices, and processing power. Network infrastructure is such a wide and vast field that volumes have been written on the subject and many schools (primarily within the computer science department) offer entire programs and majors in IT networks. Our discussion here focuses on only the five major types of network infrastructures.

DECENTRALIZED INFRASTRUCTURE A ***decentralized infrastructure*** involves little or no sharing of IT and other resources such as information. Generally, this infrastructure arises from users or departments developing their own systems or applications, without any central control (see Figure 7.6).

The decentralized infrastructure gives users the liberty to develop applications that meet their needs and maintain control over the applications they develop. However, this infrastructure generally prevents users from easily combining processing power or even comparing information between various information systems. It also encourages data duplication, frequently leading to inconsistencies. For example, Figure 7.6 illustrates a finance, a marketing, and a shipping department, each technologically isolated from the other. Decentralized infrastructures often arise in companies having a decentralized management approach and typically in companies built through acquisition. Decentralized infrastructures are almost a thing of the past.

Figure 7.7

Centralized Infrastructure



CENTRALIZED INFRASTRUCTURE A ***centralized infrastructure*** involves sharing information systems in one central area or on one central mainframe. For a long time, mainframes were the only computers available for business. By their nature, they dictated that information systems infrastructures be centralized because typically all applications and information were stored on a company's single mainframe. As an example, Figure 7.7 shows a typical layout of a centralized infrastructure. With the introduction of inexpensive desktop computers and reliable data communications technology, almost all organizations have moved away from the centralized infrastructure.

The great advantage of a centralized infrastructure is that it allows a high degree of control, making it easy to (1) maintain standards of hardware, software, procedures, and operations and (2) control access to information. The main disadvantage of a centralized infrastructure is its inflexibility. A centralized infrastructure is run so that it can be used by everyone, but that does not mean that the system is optimal for everyone. Different departments and remote sites have different information needs. As with the decentralized infrastructure, centralized infrastructures are dinosaurs from a different time.

DISTRIBUTED INFRASTRUCTURE A ***distributed infrastructure*** involves distributing the information and processing power of IT systems via a network. (This is the first true *network* infrastructure.) By connecting all the information systems via a distributed infrastructure, all locations can share information and applications (see Figure 7.8). The major benefit of this is that processing activity can be allocated to the locations(s) where it can most efficiently be done. To improve performance and reduce network traffic, a distributed infrastructure will often store the same application and/or information in two or more locations.

Distributed infrastructures are more complex than centralized infrastructures for several reasons. First, the distributed infrastructure must be able to determine the location of specific applications and information. Second, it must be more sophisticated in determining the optimal way to request the application and information. The order in which the system processes a request can make a significant difference in the amount of data transmitted over the network.

CLIENT/SERVER INFRASTRUCTURE (CLIENT/SERVER NETWORK) A ***client/server infrastructure*** (or ***client/server network***) has one or more computers that are *servers* which provide services to other computers, called *clients*. The client/server infrastructure is a form of distributed infrastructure. The basic notion of a client/server infrastructure is that application processing is divided between the client and the server. The functions of an information system are divided among connected computers (or clients) on a network while centralizing processing and storage for all information on a server. For instance, when you are surfing on the Internet, this is an example of a client/server infrastructure.

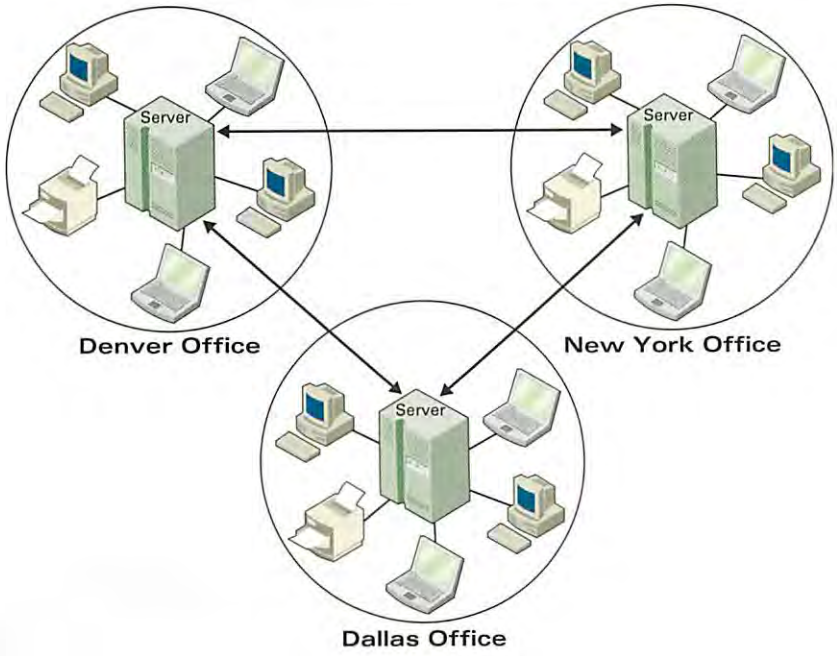


Figure 7.8
Distributed Infrastructure

INDUSTRY PERSPECTIVE

ONE VIEW FOR DEL MONTE FOODS

From its roots in the California Gold Rush era, San Francisco–headquartered Del Monte Foods has grown to become the nation's largest producer and distributor of premium quality processed fruits, vegetables, and tomato products. With annual sales of over $3 billion, Del Monte is also one of the country's largest producers, distributors, and marketers of private-label food and pet products with a powerful portfolio of brands including Del Monte, StarKist, Nature's Goodness, 9Lives, and Kibbles 'n Bits.

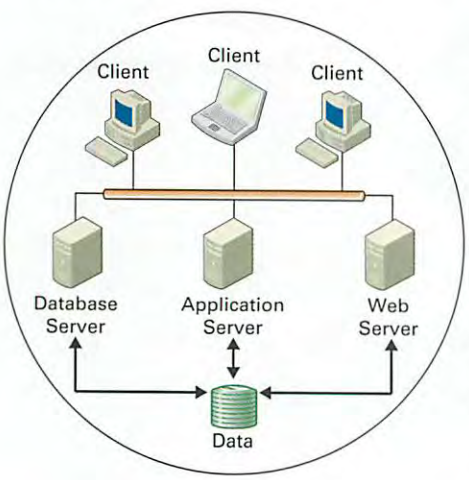
Del Monte's acquisition of certain businesses (such as StarKist, Nature's Goodness, 9Lives, and Kibbles 'n Bits) from the H. J. Heinz Company required an integration between Del Monte's and H. J. Heinz's business processes. Del Monte needed to overhaul its IT infrastructure, migrating from multiple platforms including UNIX and mainframe systems and consolidating applications centrally on a single system. The work required integration of business processes across manufacturing, financial, supply chain, decision support, and transactional reporting areas.

The revamp of Del Monte's architecture stemmed from a strategic decision. Del Monte decided to implement an ERP system to support its entire U.S. operations, headquarters in San Francisco, operations in Pittsburgh, and distribution centers and manufacturing facilities across the country. The company concluded that the only way it could unite its global operations and open its system to its customers, which are mainly large retailers, was through the use of an ERP system.

Among other key factors was the need to embrace an e-business strategy. The challenge facing Del Monte was to select an ERP system to merge multiple systems quickly and cost effectively. If financial and customer service targets were to be achieved, Del Monte needed to integrate new businesses that more than doubled the size of the company. Since implementing the ERP system, customers and trading partners are now provided with a single, consistent, and integrated view of the company.[4,5]

Figure 7.9

Client/Server Infrastructure



Typical components of this type of infrastructure include an Internet browser, a personal computer (e.g., the client), and a Web server.

The primary advantage of the client/server infrastructure is that it offloads the application programs and information from the server. However, because processing occurs at many client locations, and the client and server interact frequently and extensively, information must flow rapidly between server and clients for adequate performance (see Figure 7.9). The client/server infrastructure thereby places a heavy load on the network capacity, which can sometimes be a disadvantage. To read about how a client/server network actually works, see either *Extended Learning Module E* or *Extended Learning Module G*.

TIERED INFRASTRUCTURE Most enterprise applications are now developed using a tiered infrastructure. In a ***tiered infrastructure*** (sometimes referred to as a ***layer infrastructure***), the IT system is partitioned into tiers (or layers) where each tier (or layer) performs a specific type of functionality. The concept of a tiered infrastructure has evolved from 1-tier to n-tiers. A "tier" can be defined as "one of two or more rows, levels, or ranks arranged one above another." Figure 7.10 illustrates the concept of a tiered infrastructure.

- A ***1-tier infrastructure*** is the most basic setup because it involves a single tier on a single machine. Think of an application that runs on your PC—everything you need to run the application (data storage, business logic, user interface, and

INDUSTRY PERSPECTIVE

SOA + SAP: COCA-COLA'S NEW SECRET RECIPE

Coca-Cola's business model is a common one among well-known, almost market dominant, franchisers. Coca-Cola gets the majority of its $18 billion in annual revenue from franchise fees it earns from bottlers all over the world. Bottlers, along with the franchise, license Coke's secret recipe and many others including recipes for Odwalla, Nestea, Minute Maid, and Sprite. What Coca-Cola is now hoping is that bottlers will also buy into adopting common business practices using a service-oriented architecture ERP system.

The target platform chosen by Coca-Cola is mySAP enterprise resource planning (ERP) by SAP. If it works, Coca-Cola and its bottlers stand to make and save a lot of money, and SAP will be able to position itself as one of the dominant players in SoA-enabled ERP.

Already, Coca-Cola and many of its bottlers use versions of SAP for finance, manufacturing, and a number of administrative functions. But Coca-Cola wants everyone to move to a "services" architecture environment. Coca-Cola hopes that this services standardization will make its supply chain more efficient and reduce costs.

In explaining why a services approach is so vitally important, Jean-Michel Ares, CIO of Coca-Cola, stated, "That will allow bottlers to converge one step at a time, one process area at a time, one module at a time, at a time that's right for the bottler. We can march across the bottling world incrementally."[6]

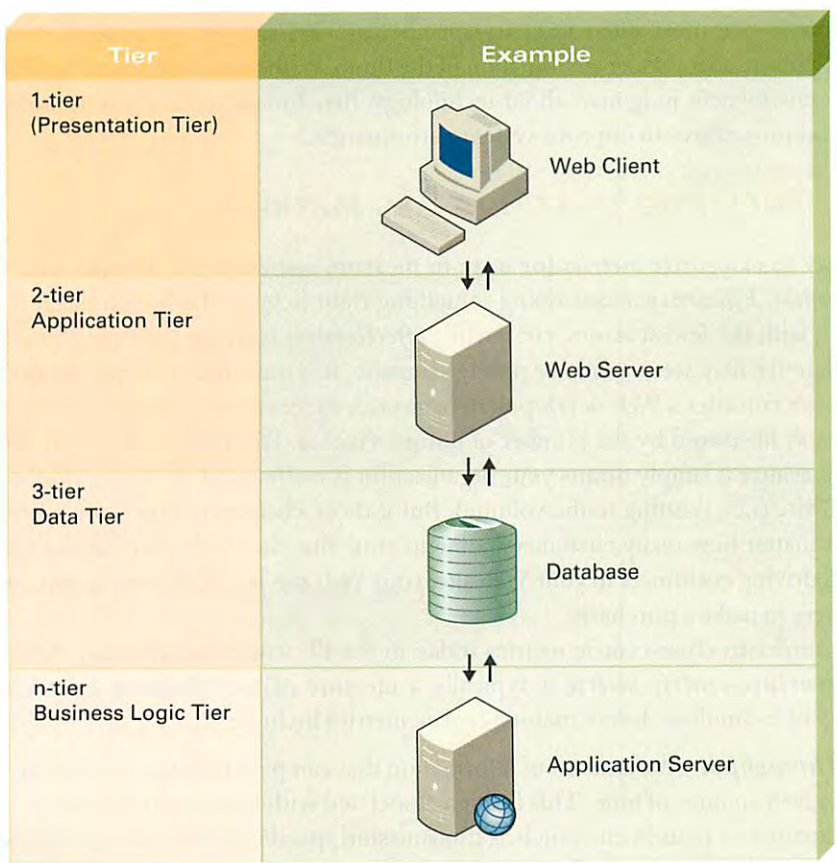


Figure 7.10

n-Tier Infrastructure Model

so forth) is wrapped up together. An example of a 1-tiered application is a basic word processor or a desktop file utility program.

- A ***2-tier infrastructure*** is the basic client/server relationship. In essence, the client handles the display, the server handles the request (e.g., database query), and the application tier is contained on one or both of the two tiers.
- A ***3-tier infrastructure*** is the most common approach used for Web applications today. A typical example of this model is the Web browser that acts as the client, an application server that handles the business logic, and a separate tier (such as a DBMS) that handles database functions.
- An ***n-tier infrastructure*** balances the work of the network over several different servers. The letter "n" stands for any number of tiers. Traditionally, an n-tier infrastructure starts with a basic 3-tier model and expands on it to allow for greater performance, scalability, and a host of other benefits.

LEARNING OUTCOME 3

IT Success Metrics

As with any organizational initiative, you must build a case for the acquisition, development, and use of technology. Technology costs money, not only for the hardware, software, and other IT-related components but also for the people involved, the changes to business processes, and the foregone opportunity to pursue other initiatives.

To justify the costs of technology, you need to be able to measure the success of technology. ***Benchmarking*** is the process of continuously measuring system results and comparing those results to ***benchmarks***—baseline values a system seeks to attain. Benchmarks are most often industry-specific, process-specific, generated internal to your organization, or some combination of the three. With these metrics in hand you can make a cost/benefit judgment about technology. Benchmarking can lead to identifying steps and procedures to improve system performance.

EFFICIENCY AND EFFECTIVENESS METRICS

Efficiency Versus Effectiveness

One way to categorize *metrics* (or ways to measure something) is by *efficiency* versus *effectiveness*. ***Efficiency*** means doing something right (e.g., in the least time, at the lowest cost, with the fewest errors, etc.), while ***effectiveness*** is doing the right thing. While the difference may seem subtle or purely semantic, it is not—the concepts are quite distinct. Let's consider a Web development project. A success metric would be Web traffic, perhaps as measured by the number of unique visitors. We refer to that as an *efficiency* metric because it simply means your organization is really good at driving customers to its Web site (i.e., creating traffic volume). But if those customers don't buy anything, it doesn't matter how many customers come to your site. So, while you may be very efficient in driving customers to your Web site, your Web site is not *effective* at getting those customers to make a purchase.

Most infrastructure-centric metrics today in the IT world are *efficiency metrics*. An ***infrastructure-centric metric*** is typically a measure of the efficiency, speed, and/or capacity of technology. Infrastructure-centric metrics include:

- ***Throughput***—the amount of information that can pass through a system in a given amount of time. This is often associated with telecommunications capabilities (bandwidth) such as transmission speeds (kilobits per second, Kbps, and megabits per second, Mbps).
- ***Transaction speed***—the speed at which a system can process a transaction.

- ***System availability***—usually measured inversely as *downtime,* or the average amount of time a system is down and unavailable to end users and customers. This does not include regularly scheduled maintenance.
- ***Accuracy***—also usually measured inversely as *error rate,* or the number of errors per thousand (or million) that a system generates. This is analogous to defects per thousand or million in manufacturing.
- ***Response time***—average time to respond to a user-generated event, such as a request for a report, a mouse click, and so on.
- ***Scalability***—how well a system can adapt to increased demands. This is more of a conceptual metric that assesses your ability to upgrade the implemented infrastructure at minimal cost and service interruption.

Effectiveness metrics, on the other hand, measure results of the technology or application in some environment. For example, call centers have numerous effectiveness (success) metrics as do Web e-business applications. We'll explore several metrics for both of those in a moment. Customer relationship management and supply chain management systems have numerous associated metrics, as well, including (but certainly not limited to):

- Customer relationship management (CRM)
  - Number of cross-selling successes
  - Cost-per-thousand (CPM)—sales dollars generated per dollar of advertising
  - Number of new customers generated
  - Average length of time a customer stays active (i.e., continues to buy products and services from you)
- Supply chain management (SCM)
  - Number of stockouts
  - Excess inventory
  - Distribution and warehousing costs

It's interesting to note in the above lists that you want to increase the value of all of the CRM metrics (e.g., the more cross-selling successes you have the better). This is a common characteristics of metrics that measure top-line initiatives. Conversely, you want to reduce the value of all the SCM metrics (e.g., have as little excess inventory as possible). This is a common characteristic of metrics that measure bottom-line initiatives.

If you consider the graph in Figure 7.11, you can begin to understand the relationship between efficiency and effectiveness metrics. Bottom-line initiatives, such as supply chain management, seek to optimize efficiency metrics while not negatively affecting

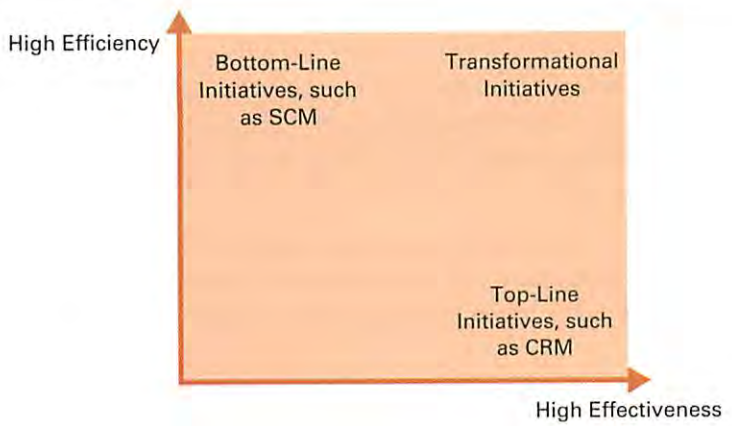


Figure 7.11
Efficiency and Effectiveness Metrics

effectiveness metrics, and top-line initiatives, such as customer relationship management, seek to optimize effectiveness metrics while not negatively affecting efficiency metrics. (It might appear in Figure 7.11 that the optimal area of operation is the upper-right area, but that's simply not true and seldom achievable except when undertaking extraordinary transformational activities. An excellent example here is that of Apple iTunes, which created both tremendous efficiency and effectiveness.)

WEB-CENTRIC METRICS

A ***Web-centric metric*** is a measure of the success of your Web and e-business initiatives. There are literally hundreds of Web-centric metrics you can use, with some being general to almost any Web or e-business initiative and others being very dependent on the particular initiative. Common Web-centric metrics include:

- ***Unique visitors***—the number of unique visitors to your sites in a given time. This is commonly used by Nielsen/Net ratings to rank the most popular Web sites.
- ***Total hits***—number of visits to your Web site, many of which may be by the same visitor.
- ***Page exposures***—average number of page exposures to an individual visitor.
- ***Conversion rate***—percentage of potential customers who visit your site who actually buy something.
- ***Click-through***—count of the number of people who visit a site, click on an ad, and are taken to the site of the advertiser.
- ***Cost-per-thousand (CPM)***—sales dollars generated per dollar of advertising. This is commonly used to make the case for spending money to appear on a search engine.
- ***Abandoned registrations***—number of visitors who start the process of completing a registration page and then abandon the activity.
- ***Abandoned shopping carts***—the number of visitors who create a shopping cart and start shopping and then abandon the activity before paying for the merchandise.

CALL CENTER METRICS

Call center metrics measure the success of call center efforts. Typical call center metrics include:

- ***Abandon rate***—the percentage number of callers who hang up while waiting for their call to be answered.
- ***Average speed to answer (ASA)***—the average time, usually in seconds, that it takes for a call to be answered by an actual person.
- ***Time service factor (TSF)***—the percentage of calls answered within a specific time frame, such as 30 or 90 seconds.
- ***First call resolution (FCR)***—the percentage of calls that can be resolved without having to call back.

If your call center operations are partly automated in the hope of helping people with common questions, then you have additional metrics such as the percentage of people who use the automated system and then request also to speak to a service representative.

FINANCIAL METRICS

Ultimately, you must make sense of all your other metrics by comparing them to or incorporating them into financial metrics, or what is more commonly referred to as *capital analysis financial models.* Organizations must be able to express the success of their IT initiatives in terms of dollars and cents. How to compute and interpret all the various financial metrics is certainly beyond the scope of this book. You will, however, explore and work with these types of metrics in your finance, accounting, and business strategies. Here, we simply define each. Figure 7.12 provides more detail.

- *Payback method*—number of years to recoup the cost of an initiative based on projected annual net cash flow.
- *Cost-benefit ratio*—a ratio comparing the benefits to the costs.
- *Return on investment (ROI)*—the overall value of an initiative (total benefits minus total costs and depreciation) as compared to the useful life of the initiative, expressed as a percentage.
- *Net present value (NPV)*—the total net present value of all cash flows over the life of an initiative.
- *Internal rate of return (IRR)*—the net present value (from above) expressed as a percentage return.

Figure 7.12

Capital Analysis Financial Models

Financial Model	Formula	Type of Result
Payback method (in years)	$\frac{\text{Original investment}}{\text{Annual net cash flow}}$	= time to pay back
Cost-benefit ratio (as a factor)	$\frac{\text{Benefits}}{\text{Costs}}$	= cost-benefit ratio
Return on investment (ROI, as a percent)	$\frac{\text{(Benefits – cost – depreciation)}}{\text{useful system life}}$	= net benefit
Net present value (NPV, in $)	Total net present value of all cash flows*	= value today of return in future
Internal rate of return (IRR, as a percent)	The NPV represented as a percentage return*	= expected return

*The net present value and internal rate of return calculation are too complex to summarize above. So both are defined completely below:

Net present value = sum of the present value of all future payments less the initial cost
$= -CF_0 + \Sigma[CF_t/(1 + r)^t]$ where CF_0 = the initial cost
CF_t = each future payment
r = the discount rate
t = the number of the time payment

Internal rate of return = the rate that completes the following summation equation
Cost $= \Sigma[CF_t/(1 + IRR)^t]$ where CF_t = the future payments
IRR = the internal rate of return
t = the number of the payment

GLOBAL PERSPECTIVE

THE LONDON BRIDGE MAY FALL DOWN BUT THE EXCHANGE WON'T

The London Stock Exchange is among the most admired equity exchanges in the world, not only for its long-standing position in the financial community but also for its technology deployment and infrastructure. And when you process 15 million real-time messages per day (with peaks of 2,000 messages per second), your infrastructure had better be good.

In choosing a new technology infrastructure on which to ride into the future, the London Stock Exchange focused a great deal of its efforts on requiring vendors to meet benchmarks across a broad range of metrics. Below are just a few.

- *Guaranteed performance*—in financial trading, information is valuable only if it reaches traders within the first second.
- *Development costs*—reduced development costs and development cycle times means more productivity. Ian Homan, head of Technology for the London Exchange, estimates that implementation of the new infrastructure occurred in one-fifth to one-third the time it would have taken to implement other vendor infrastructures.
- *Scalability*—according to David Lester, CIO for the London Stock Exchange, "We want to be able to extend it and make it richer. Investment decisions of this kind aren't made carelessly and the ability to scale to our future needs was a critical factor."[7,8]

SERVICE LEVEL AGREEMENTS REVISITED

To close our discussion of metrics, let's revisit the service level agreement. As we defined it in Chapter 6, a ***service level agreement (SLA)*** is a formal, contractually obligated agreement between two parties. We further described an SLA by saying that, within different environments, it takes on different meanings.

An important aspect of any SLA is a definition of the metrics that will be used to measure the success of the interactions between the two parties. Suppose, for example, that your organization decided to outsource its call center operations. The SLA would cover in great detail the financial agreement, such as how much money your organization will pay the call center outsourcing organization. It would also include key metrics and benchmarks that the outsourcing organization must meet to stay in compliance with the SLA. These metrics would include many of those we listed above including average speed to answer, abandon rate, and time service factor.

Many organizations today also outsource their Web development and e-business activities. In that case, your SLA would include key metrics and benchmarks around such areas as those we discussed within the context of Web-centric metrics. The definitions of these metrics and benchmarks are usually not present in the SLA, which is more business oriented and void of detailed technical specifications. They would, instead, appear in a supporting document (similar to a contract addendum) called a ***service level specification (SLS)*** or ***service level objective (SLO)***.

Another important business relationship in which you find the use of a service level agreement is that between an organization and an application service provider. An ***application service provider (ASP)*** supplies software applications (and often related services such as maintenance, technical support, information storage, and the like) over the Internet that would otherwise reside on customers' computers. The ASP model is very popular with businesses. For example, many businesses use CRM software provided by

Salesforce.com, which hosts its CRM software on its Internet servers. Businesses subscribing to Salesforce.com's CRM software access it over the Internet, thus eliminating the need to buy, store, and maintain the software on their own computers. A service level agreement is vitally important and includes application-specific metrics (such as those associated with CRM software if you use Salesforce.com's CRM software) and infrastructure-centric metrics such as system availability, transaction speed, response time, and throughput.

Business Continuity Planning

LEARNING OUTCOME 4

Business continuity planning (BCP) is a rigorous and well-informed organizational methodology for developing a ***business continuity plan,*** a step-by-step guideline defining how the organization will recover from a disaster or extended disruption of its business processes. In past years, with respect to IT systems and information, the business continuity plan went by other names, *disaster recovery plan* and *contingency plan.* Given the number of natural disasters and terrorist attacks worldwide, along with businesses' increasing dependency on all their processes and resources (not just IT and information), however, the general trend has been to develop a more all-encompassing *business continuity plan* that includes all aspects of the organization.

Business Continuity Planning

The BCP methodology looks very similar to the systems development life cycle (see Figure 7.13). It starts with the organization's strategic plan and moves through various phases including analysis, design, implementation, testing, and maintenance. We'll focus our discussion of BCP on information technology and IT-related issues.

PHASE 1: ORGANIZATIONAL STRATEGIC PLAN

Business continuity planning starts with your organization's strategic plan, which informs you of the relative importance of resources, processes, systems, and other organizational assets. It's important to understand and develop a ranking of the importance of these assets because you cannot (and should not) develop a business continuity plan that enables you to recover every asset within minutes of some sort of disaster. That would be prohibitively expensive and unnecessary. Your organization can afford to live without some assets (i.e., systems and information) for several days or even weeks. Payroll software may be an example here. For other assets, such as customer ordering and supply chain applications, it may critical that your organization gets those up and running with

Figure 7.13
Business Continuity Planning (BCP)

minimal interruption. Data centers are also typically identified as high priorities for most organizations. According to CPM Research, "improved business continuity" was cited 70 percent of the time by respondents as being a key data center issue.[9]

PHASE 2: ANALYSIS

In the BCP analysis phase, you perform impact analysis, threat analysis, and impact scenario analysis, and then build a requirement recovery document.

- (1) *Impact analysis*—Here you seek to truly differentiate between critical, core IT applications and information and those that are noncritical. Key factors supporting your analysis include: the financial impact to the organization for the loss of IT applications and information over time, implications for stakeholders (e.g., customer loss of power if you provide utilities), and cost estimates of recovery. Impact analysis is often called ***risk assessment,*** the process of evaluating IT assets, their importance of the organization, and their susceptibility to threats.
- (2) *Threat analysis*—In step 2 of BCP analysis, you document the possible threats to your organization and its assets. These can and often do include disease, earthquakes (depending on the geographical location of your organization), fire, flood, cyber attack, terrorism, and utility outages. An assessment of these helps you develop an understanding of the magnitude of threats and how you should choose to recover from them. Note that in the case of e-business activities threats can include greatly increased shopping traffic. For example, on the day after Thanksgiving in 2006 Wal-Mart's Web site went down because so many people visited its site in the hope of buying the hard-to-find T.M.X. Elmo.[10]
- (3) *Impact scenario analysis*—In step 3, you consider each threat (from step 2) and build a worst-case scenario for each (as opposed to smaller impact scenarios such as 10 percent of the workforce out due to a flu outbreak). An impact scenario analysis provides further definition and detail concerning the scope and magnitude of each possible disaster.
- (4) *Requirement recovery document*—Armed with the information from steps 1 through 3, you finally build a ***requirement recovery document,*** a detailed document which describes (a) the distinction between critical and noncritical IT systems and information, (b) each possible threat, and (c) the possible worst-case scenarios that can result from each disaster. This document becomes the basis for the design phase which follows.

PHASE 3: DESIGN

Using the requirement recovery document, in the design phase, you design a formal, technical, and detailed plan for recovering from a disaster—a ***disaster recovery plan.*** A good disaster recovery plan takes into consideration the location of the backup information. Many organizations choose to store backup information in an off-site storage facility, or a place that is located separate from the company and often owned by another company, such as a collocation facility. A ***collocation facility*** is available to a company that rents space and telecommunications equipment from another company. One such company is StorageTek, which specializes in providing off-site data storage and disaster recovery solutions.

A good disaster recovery plan also considers the actual facility where employees will work. A ***hot site*** is a separate and fully equipped facility where the company can move immediately after a disaster and resume business. A ***cold site*** is a separate facility that does not have any computer equipment but is a place where employees can move after the disaster.

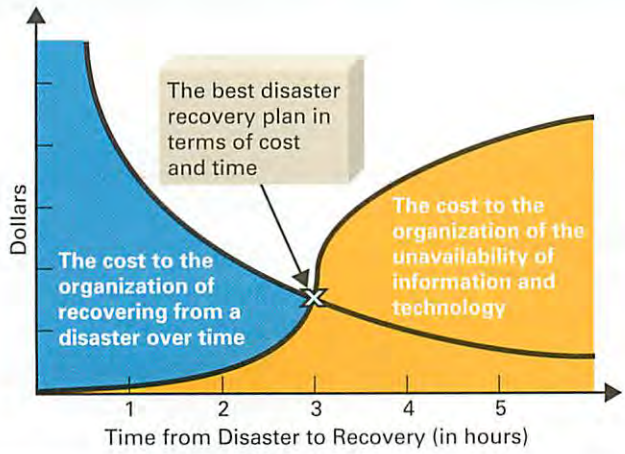


Figure 7.14

Deciding How Much to Spend on Disaster Recovery

A disaster recovery plan should be based on a *disaster recovery cost curve* (Figure 7.14). A ***disaster recovery cost curve*** charts (1) the cost to your organization of the unavailability of information and technology and (2) the cost to your organization of recovering from a disaster over time. Where the two curves intersect is the best recovery plan in terms of cost and time. Being able to restore information and IT systems quickly in the event of a disaster is obviously a crucial aspect of an IT infrastructure.

PHASE 4: IMPLEMENTATION

At this point, business continuity planning diverges somewhat from the SDLC. In the SDLC, you would develop and test the solution before implementing it. In business continuity planning, you must begin to implement your disaster recovery plan before testing it. That is, you need to engage any businesses that will be providing collocation facilities, hot sites, and/or cold sites, and implement the necessary procedures for recovering from a disaster. You train your employees concerning what to do in case of any of the disasters. You also evaluate each IT system and ensure that it is configured optimally for recovering from a disaster. You can now test the disaster recovery plan.

PHASE 5: TESTING

Testing in business continuity planning involves executing simulated scenarios of disasters and having employees execute on the disaster recovery plan to ensure that the solution satisfies your organization's recovery requirements. If noticeable deficiencies are identified, your organization should return to steps 3 and 4 (design and implementation) and reconfigure and reimplement the disaster recovery plan accordingly.

This sort of testing does not stop once you believe you have developed an optimal plan. The environment in which your organization operates changes almost daily. You should test your disaster recovery plan on at least an annual basis, and more realistically several times a year.

PHASE 6: MAINTENANCE

Finally, you need to continually assess new threats and reevaluate your IT systems and related assets to determine their changing importance to the organization. As with the SDLC, no "system" is ever complete; it needs constant monitoring, support, and maintenance.

INDUSTRY PERSPECTIVE

ENTERGY SURVIVES KATRINA

Based in New Orleans, Entergy Corp. employs 14,500 people to provide electricity to 2.6 million utility consumers in Arkansas, Louisiana, Texas, and Mississippi. In late August 2005, Hurricane Katrina tested the resolve of all of Entergy's employees and the preparedness of its disaster recovery plan.

Employees rose to the occasion and the disaster recovery plan was executed well. Although returning all utilities to every customer in the region took several months (because of factors well beyond Entergy's control), it was Entergy's dedication to building a successful disaster recovery plan that paid dividends.

Located in the Gulf of Mexico in the heart of hurricane country in the southern part of the United States, the energy company had an effective disaster recovery plan well in place. Entergy tested its disaster recovery plan at least once a year. The previous year that test had come in the form of preparing for Hurricane Ivan, which ultimately made landfall to the east in Florida. Just five months before Hurricane Katrina hit, the company conducted a massive storm drill which simulated what everyone thought was the worst-case scenario—a major hurricane followed by extreme flooding.

Entergy's disaster recovery plan included several key elements, including:

- Start the execution of the disaster recovery plan at least 72 hours in advance of the predicted landfall of a hurricane.
- Activate Entergy's disaster recovery site in Little Rock, Arkansas.
- Activate Entergy's storm command center (also called *The Power House*) in Jackson, Mississippi.
- Still in advance of the predicted landfall of a hurricane, prepare the company's systems that would be most critical in restoring electricity to utility consumers.

According to everyone, the disaster recovery plan worked well, but Ray Johnson, CIO of Entergy, thought some changes needed to be made. As he explained it, "We never follow the plan to the letter. In the IT space, the plan is very solid in terms of what we have to do. But we're always working—our core IT staff in conjunction with representatives from the business areas—to see if we need to change priorities."[11]

Student Learning Outcomes Revisited

1. **Describe how a service-oriented architecture can be used as a philosophical approach to help the organization of the future meet all its IT-related needs.** A service-oriented architecture can be applied to help your organization respond to:
   - *Customers*—through multichannel service delivery and the provision of customizable products and services.
   - *End users*—through fully integrated ERP systems supporting interoperability and mobile computing.
   - *Software development*—as a framework for supporting development methodologies such as RAD, XP, and agile that lead to exciting new deployments such as Web 2.0.
   - *Information needs*—including access to all types of integrated information (and business intelligence and knowledge), standard information formats, integrity controls, and the elimination of redundant information.

- *Hardware requirements*—through the integration of different technologies, providing large storage capacities, and maintaining safe and secure telecommunications platforms.

2. **Define and describe the various hardware and software infrastructure considerations in an organization.** Hardware and software infrastructure considerations include:
   - Enterprise resource planning (ERP) systems provide interoperability within an ERP vendor and among modules of different ERP vendors and also hide the underlying IT infrastructure of information and hardware from end users and customers.
   - Network infrastructures include: ***Decentralized***—little or no sharing of IT and other resources such as information; ***Centralized***—sharing of information systems in one central area or one central mainframe; ***Distributed***—distributing the information and processing power of IT systems via a network; ***Client/server***—one or more computers that are servers which provide services to other computers, called clients; ***Tiered (layer)***—the IT system is partitioned into tiers where each tier performs a specific type of functionality.
3. **Compare and contrast commonly used metrics for assessing the success of IT systems and IT-related initiatives.** Metrics are simply ways to measure something. Common IT metrics can be categorized as:
   - ***Infrastructure-centric metrics***—Measures of efficiency, speed, and/or capacity of technology, including ***throughput, transaction speed, system availability, accuracy, response time,*** and ***scalability.***
   - ***Web-centric metrics***—Measures of the success of your Web and e-business initiatives, including ***unique visitors, total hits, page exposures, conversion rate, click-through, cost-per-thousand (CPM), abandoned registrations,*** and ***abandoned shopping carts.***
   - ***Call center metrics***—Measures of the success of call center efforts, including ***abandon rate, average speed to answer (ASA), time service factor (TSF),*** and ***first call resolution (FCR).***
   - ***Financial metrics*** - Also called ***capital analysis financial models,*** including ***payback method, cost-benefit ratio, return on investment (ROI), net present value (NPV),*** and ***internal rate of return (IRR).***
4. **Describe business continuity planning (BCP) and its phases.** ***Business continuity planning (BCP)*** is a rigorous and well-informed organizational methodology for developing a ***business continuity plan,*** a step-by-step guideline defining how the organization will recover from a disaster or extended disruption of its business processes. The phases of business continuity planning include:
   - *Phase 1: Organizational strategic plan*—it all starts here with understanding the relative importance of resources, systems, processes, and other organizational assets.
   - *Phase 2: Analysis*—perform impact analysis, threat analysis, and impact scenario analysis and build a ***requirement recovery document,*** a detailed document that describes (1) the distinction between critical and noncritical IT systems and information, (2) each possible threat, and (3) the possible worst-case scenarios that can result from each disaster.
   - *Phase 3: Design*—using the requirement recovery document, create a ***disaster recovery plan,*** which identifies collocation facilities, hot sites, and cold sites and illustrates a disaster recovery cost curve (the cost to your organization of the unavailability of information and technology as compared to the cost to your organization of recovering from a disaster over time).
   - *Phase 4: Implementation*—engage businesses that will provide collocation facilities, hot sites, and cold sites; implement necessary procedures for recovering from a disaster; train employees; evaluate each IT system to ensure its configuration is optimal for recovering from a disaster.
   - *Phase 5: Testing*—executing simulated scenarios of disasters and having employees execute on the disaster recovery plan.
   - *Phase 6: Maintenance*—continually assess new threats and reevaluate your IT systems and related assets to determine their changing importance to the organization.

CLOSING CASE STUDY ONE

INTERNATIONAL TRUCK MAKES A HUGE BET ON A SERVICE-ORIENTED ARCHITECTURE

Service-oriented architectures (SoAs) do seem to be the future for integrated IT systems within an organization, bringing together both applications and information in a seamless fashion. But SoAs are still in their infancy, and some companies don't want to wait until they mature. They are willing to bet on a service-oriented architecture right now. One such company is International Truck and Engine Corporation.

International Truck's SoA resulted from needing more timely information so it could identify assembly-plant problems sooner. Its current IT systems, all legacy information systems, didn't share information easily, and that led to bottlenecks in production, excessive defects and returns, shortfalls in inventory, and a host of other problems that can spell doom for a manufacturing-intensive company like International Truck.

So, the company forged ahead in embracing an SoA, knowing that it would be able to buy many of the software components that it needed and that others would have to be written from scratch by in-house IT specialists. The first focus for International Truck was the Common Vehicle Tracking system, a system that tracks production according to a specific vehicle or vehicle type. International truck produces everything from RV motor homes to military transports.

The Common Vehicle Tracking system was a high-profile, high-payoff project with a tight deadline. The company expected the system to save it at least $3 million annually. The system, now complete and installed in one factory, tracks in near real time all information relating to works-in-progress and finished inventory. Prior to its implementation, Art Data, Vice President of Information Technology at International Truck, succinctly explained, "We weren't doing it very well." International Truck's legacy systems stored isolated data and information in applications such as computer-aided manufacturing, in-house developed order management, and even commercial ERP software.

The new system uses a combination of many different types of software, one of the primary goals of a service-oriented architecture. For example, it uses a data integration tool from SSA Global to extract information from International Truck's Baan ERP system. From there, in-house developed software bridges to the order management system.

International Truck is also working on an SoA extension to interface with the systems of its dealers. The company already has a centralized server that its 400 dealers use to access parts catalogs and sales tools. However, dealers can choose their own internal dealer management software systems. This makes communicating information in a common format problematic, at best. International Truck is currently working with automotive industry software vendors to create common services (i.e., software modules) that will communicate information in a standard format.

In the future, International Truck will even extend its SoA architecture perspective into the vehicles themselves. Using the vehicles' electronics system, GPS, and cellular technology, owners of the trucks will be able to track the location of their vehicles.[12]

Questions

1. With respect to its customers (dealers in this case), how is International Truck using a service-oriented architecture to meet their needs? How does using an SoA further help International Truck erect an entry barrier (from Chapter 1)?
2. With respect to information needs, what advantages and efficiencies is International Truck hoping to gain by using a service-oriented architecture? Why are these advantages and efficiencies not possible with its current legacy systems?
3. Recalling our discussion of software development methodologies in Chapter 6 that focus on component-based development (i.e., RAD, XP, and agile), how is International Truck able to integrate new software modules with existing ones?

4. What key infrastructure-centric metrics could International Truck use to justify its movement toward a service-oriented architecture? For each that you identify, provide a short description of why the metric is important.

5. How important is it for International Truck to have a good business continuity plan in place? What key IT systems and other IT resources do you believe would be at the top of the list for quick recovery? Why?

CLOSING CASE STUDY TWO

ROGER WILLIAMS MEDICAL CENTER BOOSTS ITS IT INFRASTRUCTURE

Roger Williams Medical Center is a 220-bed hospital located in Providence, Rhode Island. It has been widely recognized as a leading provider of health care, not only on the East coast but throughout the United States. Roger Williams faced a difficult challenge implementing several key new IT systems because of its aging IT infrastructure.

According to Kevin Frederick, Director of Technology, at Roger Williams, "Our legacy systems had reached the end of their life cycle, and we knew that we would need a big jump in processing power and storage capacity to support our new applications." Those new applications included MEDITECH, a new comprehensive patient-care system, and the Picture Archive Communication System (PACS). PACS stores and distributes electronic images from CAT scans, MRI, ultrasound and other image-based tests (totaling more than 19,000 annually), allowing health care providers to diagnose conditions and diseases using high-resolution monitors.

To achieve the scalability, availability, and performance required, Roger Williams contracted HP Services to design, develop, deliver, and support its needed IT infrastructure. That infrastructure includes a server farm of 41 servers connected to a storage area network with 4 terabytes of capacity to start.

Kevin made the following statements regarding working with HP Services and its IT infrastructure: "The BladeServer systems [server farm] enabled us to reduce the footprint of our racked servers by 50 percent, while giving us plenty of headroom for growth . . . The blades have a smart design—with power integrated into the chassis, they are easy to deploy and fast to swap. If a problem were to develop, we can easily deploy one of our 'cold spares' and minimize any disruption of service. HP Services spent the week assembling the SANS and educated us along the way. I couldn't have asked for it to go any better." Even users of the new system frequently comment on how much faster the new system performs.

The IT infrastructure certainly supports scalability and increased processing capacity. After the new system had been online for nine months, Roger Williams added 11 more servers as it expanded the user-facing applications within MEDITECH and PACS. Many of those user-facing applications are critical to Roger Williams' success. These include numerous clinical modules, Patient Tracker (for emergency room services), and Long Term Care (for nursing home support).

The future of Roger Williams and its IT infrastructure is to develop an intensive business continuity plan that will provide for redundant data servers.[13,14]

Questions

1. In this chapter, we noted that many effectiveness metrics depend on the application under scrutiny, such as supply chain management and customer relationship management. What sort of effectiveness metrics would the health care industry likely use to evaluate the success of IT systems?

Where possible, highlight the use of these effectiveness metrics by Roger Williams.

2. In the final part of this case study, Roger Williams was making plans to create a business continuity plan for its information. Why do you think the hospital had not already developed such a plan in the event of a disaster? How is HP's new IT infrastructure going to help Roger Williams build a robust business continuity plan?
3. According to the network infrastructures we discussed in this chapter, how would you characterize Roger Williams' new IT infrastructure? Justify your answer.
4. As a part of a business continuity plan, many businesses identify cold sites or hot sites to which they can move to carry on their business activities. What about a hospital like Roger Williams? What if it suffered a disaster such as a flood or hurricane? Where could it move to provide quality patient care?
5. What infrastructure-centric metrics can you find in this case study? Why are they so important to Roger Williams? What other infrastructure-centric metrics might be important to the hospital?

Key Terms and Concepts

1-tier infrastructure, 320
2-tier infrastructure, 322
3-tier infrastructure, 322
Abandon rate, 324
Abandoned registration, 324
Abandoned shopping cart, 324
Accuracy, 323
Application service provider (ASP), 326
Average speed to answer (ASA), 324
Benchmark, 322
Benchmarking, 322
Business continuity plan, 327
Business continuity planning (BCP), 327
Call center metric, 324
Centralized infrastructure, 318
Click-through, 324
Client/server infrastructure (client/server network), 319
Cold site, 328
Collocation facility, 328
Conversion rate, 324
Cost-per-thousand (CPM), 324
Decentralized infrastructure, 318
Disaster recovery cost curve, 329
Disaster recovery plan, 328
Distributed infrastructure, 319
Effectiveness, 322
Efficiency, 322
First call resolution (FCR), 324
Hot site, 328
Infrastructure, 313
Infrastructure-centric metric, 322
Interoperability, 316
n-tier infrastructure, 322
Page exposures, 324
Requirement recovery document, 328
Response time, 323
Risk assessment, 328
Scalability, 323
Service level agreement (SLA), 326
Service level objective (SLO), 326
Service level specification (SLS), 326
Service-oriented architecture (SoA or SOA), 310
System availability, 323
Throughput, 322
Tiered infrastructure (layer infrastructure), 320
Time service factor (TSF), 324
Total hits, 324
Transaction speed, 322
Unique visitors, 324
Web-centric metric, 324

Short-Answer Questions

1. How can a service-oriented architecture (SoA) be used to guide the organization of the future?
2. How have ERP systems evolved over the last 30 years?
3. Why is interoperability important?
4. What are the main differences between a decentralized infrastructure and a centralized infrastructure?
5. How does a client/server infrastructure work?
6. What are the four types of a tiered infrastructure?

7. How do efficiency and effectiveness metrics differ?
8. What are some commonly used infrastructure-centric metrics?
9. What are some commonly used Web-centric metrics?
10. Why are service level agreements important when contracting the services of an application service provider (ASP)?
11. What is a business continuity plan?
12. Why do organizations implement a disaster recovery plan before testing it?

Assignments and Exercises

1. **SECURITY METRICS** In this chapter, we focused on metrics for measuring the success of your IT systems including infrastructure-centric metrics, Web-centric metrics, call center metrics, and financial metrics. Another important area of metrics is security metrics, or how well you are doing at stopping viruses from coming in, protecting against identify theft, and the like. Do some research on the Web and develop a list of commonly used metrics in the area of security. Be sure to define each metric.
2. **CREATING A CAMPUS IT INFRASTRUCTURE** You have been assigned the role of student IT infrastructure manager. Your first assignment is to approve the designs for the new on-campus Internet infrastructure. You're having a meeting at 9:00 A.M. tomorrow morning to review the designs with the student IT employees. To prepare for the meeting, you must understand the student requirements and their current use of the Internet, along with future requirements. The following is a list of questions you must answer before attending the meeting. Provide an answer to each question.
   - Do you need to have a disaster recovery plan? If so what might it include?
   - Does the system require backup equipment?
   - When will the system need to be available to the students?
   - What types of access levels will the students need?
   - How will you ensure the system is reliable?
   - How will you build scalability into the system?
   - What are the minimum performance requirements for the system?
   - How will the system handle future growth?
3. **EFFICIENCY AND EFFECTIVENESS METRICS** Choose any of the Perspective boxes in this chapter or the opening case and identify and describe at least seven metrics that could be used to measure the success of the IT systems in your chosen example. For each metric, categorize it as either an efficiency or effectiveness metric. Justify your categorizations.
4. **APPLICATION SERVICE PROVIDERS** There are numerous application service providers that enable organizations to access and use Web-based application software. In the chapter we identified Salesforce.com as one such ASP. Search the Web and find at least five other ASPs. What are the names of the companies? What application software do they provide over the Web? What additional services do they provide, if any?
5. **PERSONAL BENCHMARKS AND BENCHMARKING** How do you use benchmarks and benchmarking in your personal life? Think about grades, making money, supporting charities, and the like. Choose one significant way in which you use benchmarks and benchmarking in your personal life and describe it. What are your benchmark values? How were you able to derive your benchmark values? Where did they come from?

Discussion Questions

1. On page 324, we listed and defined numerous Web-centric metrics. Which of those are efficiency metrics and which of those are effectiveness metrics? For each, provide justification for your answer and an illustration using a real-life or hypothetical Web business example.
2. IT infrastructures often mimic organizational hierarchies and the philosophical approach organizations take to the placement of the IT function (see Chapter 2). Describe which common network infrastructure (decentralized, centralized, distributed, client/server, or tiered) would best support the three structural placements of the IT function we listed on page 77.
3. What type of IT infrastructure does your school have? If it uses a client/server infrastructure how does your school's client/server network increase student productivity? What recommendations, based on the contents of this chapter, could you recommend to the IT people who manage the infrastructure?
4. How is the concept of interoperability an important aspect that you rely on in your daily life? Think about the many devices, appliances, modes of transportation, and so on that you use every day. Which of these support the concept of interoperability? How would your life change if they didn't support interoperability?
5. Many people say that efficiency and effectiveness metrics are interrelated and that you can't really have one without the other or that no organization can truly be successful without both. How are efficiency and effectiveness metrics interrelated? Must you succeed in one set before addressing the other? If so, which is first and why?
6. Consider an e-business like Amazon.com. Which Web-centric metrics on page 324 would be most important to it? Justify your answers. Now consider a content provider like CNN (www.cnn.com). Which Web-centric metrics would be most important to it? Justify your answers. Why would two e-businesses have such a different focus on Web-centric metrics?

CHAPTER PROJECTS

Group Projects

- Assessing the Value of Outsourcing Information Technology: Creating Forecasts (p. 472)
- Creating a Decision Support System: Buy versus Lease (p. 476)
- Developing an Enterprise Resource Planning System: Planning, Reporting, and Data Processing (p. 477)
- Evaluating the Next Generation: Dot-Com ASPs (p. 479)
- Evaluating the Security of Information: Wireless Network Vulnerability (p. 486)

e-Commerce Projects

- Best in computer statistics and resources (p. 488)
- Meta data (p. 489)
- Finding hosting services (p. 493)
- Searching for shareware and freeware (p. 498)