After reading this chapter, you will be able to:

- Explain the process that should be followed in building an e-commerce Web site.
- Describe the major issues surrounding the decision to outsource site development and/or hosting.
- Identify and understand the major considerations involved in choosing Web server and e-commerce merchant server software.
- Understand the issues involved in choosing the most appropriate hardware for an e-commerce site.
- Identify additional tools that can improve Web site performance.
Let’s say you’ve decided to open a Web site for your successful garden equipment company. You’ve been in business for twelve years, have established a regional brand for high-quality gardening tools, and have about 12,000 retail customers and 21 wholesale dealers who purchase from you. Based on a marketing report you commissioned, you expect that your Web site in the first year will have about 500,000 visitors, about 1,400 a day. The average visitor will look at eight pages, producing about 4 million page hits a year. About 10% will purchase something, and the rest will browse to explore prices and products. However, in peak times (during the months of April, May, June, and December), you expect peak loads of 3,000 customers a day, concentrated during the hours of 9 A.M. to 5 P.M., producing about 375 visitors per hour or 6 per second. During this time, your Web site will have to serve up about 40 screens per second with most of the content being read from a database of product and price information. Pages must be served up within 2 seconds of a customer click during peak times or customers may lose patience and go elsewhere.
Before you can proceed, there are some questions you will need answered. How many Web servers will your site require? How many CPUs should each server have? How powerful does the site’s database server need to be? What kind of connection speed do you need to the Internet? Until recently, finding the answers to questions such as these was often done on a trial-and-error basis. However, hardware and software vendors such as IBM, Microsoft, and Hewlett Packard have recently developed a number of simulation tools that can help you find the right answers.

IBM’s simulator is called the “High Volume Web Site Simulator” (HVWS Simulator) and uses a queuing model that estimates the performance and capacity of a Web server based on workload patterns, performance objectives, and specific hardware and software. The system has a very easy-to-use interface that allows users to define their business; it generates reports that allow users to assess the adequacy of their current hardware and software, and to forecast performance and identify bottlenecks that might develop by changing their hardware and software in the future.

eBay.com is one high-profile user of the HVWS Simulator. With 30 million customers, eBay’s original Web site hardware and software was slowing down to the point where it created customer resistance. While in the early years, eBay might need to serve up 1 million pages per hour, by 2002, it was facing demand for 3 to 4 million pages per hour, and nearly all of these pages were dynamic pages in the sense that their contents had to be looked up in a database and then displayed. After running the simulation of its current and likely future workload out to 2005, eBay decided to build its new trading system around IBM’s WebSphere application that integrates a variety of software tools into an integrated Web site design.
In Chapter 3, we examined the Internet and Web's infrastructure, e-commerce's technological foundation. Now it's time to take the next step: building an e-commerce site.

In this chapter, we will describe the important factors that a manager needs to consider when building an e-commerce site. Our focus will be on the managerial business decisions you must make. As you will see, building a sophisticated e-commerce site isn't easy; in fact, it can be so complicated that some companies are basing their entire business model on building and running e-commerce sites for others.

4.1 BUILDING AN E-COMMERCE WEB SITE: A SYSTEMATIC APPROACH

Building a successful e-commerce site is a complex endeavor that requires a keen understanding of business, technology, and social issues, as well as a systematic approach. In many firms today, e-commerce is just too important to be left totally to technologists and programmers.

The two most important management challenges in building a successful e-commerce site are (1) developing a clear understanding of your business objectives and (2) knowing how to choose the right technology to achieve those objectives. The first challenge requires you to build a plan for developing your firm's site. The second challenge requires you to understand some of the basic elements of e-commerce infrastructure.

Even if you decide to outsource the entire e-commerce site development and operation to a service provider, you will still need to have a site development plan and some understanding of the basic e-commerce infrastructure issues such as cost, capability, and constraints. Without a plan and a knowledge base, you will not be able to make sound management decisions about e-commerce within your firm (Laudon and Laudon, 2004).

PIECES OF THE SITE-BUILDING PUZZLE

Let's assume you are a manager for a medium-sized, industrial parts firm of around 10,000 employees worldwide, operating in ten countries in Europe, Asia, and North America. Senior management has given you a budget of $1 million to build an e-commerce site within one year. The purpose of this site will be to sell and service the firm's 20,000 customers, who are mostly small machine and metal fabricating shops around the world. Where do you start?

First, you must be aware of the main areas where you will need to make decisions (see Figure 4.1). On the organizational and human resources front, you will have to
bring together a team of individuals who possess the skill sets needed to build and manage a successful e-commerce site. This team will make the key decisions about technology, site design, and the social and information policies that will be applied at your site. The entire site development effort must be closely managed if you hope to avoid the disasters that have occurred at some firms.

You will also need to make decisions about your site’s hardware, software, and telecommunications infrastructure. While you will have technical advisors help you make these decisions, ultimately the operation of the site is your responsibility. The demands of your customers should drive your choices of technology. Your customers will want technology that enables them to find what they want easily, view the product, purchase the product, and then receive the product from your warehouses quickly. You will also have to carefully consider your site’s design. Once you have identified the key decision areas, you will need to think about a plan for the project.

**PLANNING: THE SYSTEMS DEVELOPMENT LIFE CYCLE**

Your second step in building an e-commerce site will be creating a plan document. In order to tackle a complex problem such as building an e-commerce site, you will have to proceed systematically through a series of steps. One methodology for developing an e-commerce site plan is the systems development life cycle (see Figure 4.2).

The *systems development life cycle (SDLC)* is a methodology for understanding the business objectives of any system and designing an appropriate solution. Adopting a life cycle methodology does not guarantee success, but it is far better than having no plan at all. The SDLC method also helps in creating documents that communicate to senior management the objectives of the site, important milestones, and
The five major steps involved in the systems development life cycle for an e-commerce site are:

- Systems Analysis/Planning
- Systems Design
- Building the System
- Testing
- Implementation

**SYSTEMS ANALYSIS/PLANNING: IDENTIFY BUSINESS OBJECTIVES, SYSTEM FUNCTIONALITY, AND INFORMATION REQUIREMENTS**

The systems analysis/planning step of the SDLC tries to answer the question, “What do we want the e-commerce site to do?” We will assume here that you have identified a business strategy and chosen a business model to achieve your strategic objectives (see Chapter 2). But how do you translate your strategies, business models, and ideas into a working e-commerce site?

One way to start is to identify the specific business objectives for your site, and then develop a list of system functionalities and information requirements. **Business objectives** are simply a list of capabilities you want your site to have.
System functionalities are a list of the types of information systems capabilities you will need to achieve your business objectives. The information requirements for a system are the information elements that the system must produce in order to achieve the business objectives. You will need to provide these lists to system developers and programmers so they know what you as the manager expect them to do.

Table 4.1 describes some basic business objectives, system functionalities, and information requirements for a typical e-commerce site. As shown in the table, there are nine basic business objectives that an e-commerce site must deliver. These objectives must be translated into a description of system functionalities and ultimately into a set of precise information requirements. The specific information requirements for a system typically are defined in much greater detail than Table 4.1 indicates. To a large extent, the business objectives of an e-commerce site are not that different

### Table 4.1: System Analysis: Business Objectives, System Functionality, and Information Requirements for a Typical E-Commerce Site

<table>
<thead>
<tr>
<th>Business Objective</th>
<th>System Functionality</th>
<th>Information Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display goods</td>
<td>Digital catalog</td>
<td>Dynamic text and graphics catalog</td>
</tr>
<tr>
<td>Provide product information (content)</td>
<td>Product database</td>
<td>Product description, stocking numbers, inventory levels</td>
</tr>
<tr>
<td>Personalize/customize product</td>
<td>Customer on-site tracking</td>
<td>Site log for every customer visit; data mining capability to identify common customer paths and appropriate responses</td>
</tr>
<tr>
<td>Execute a transaction</td>
<td>Shopping cart/payment system</td>
<td>Secure credit card clearing; multiple payment options</td>
</tr>
<tr>
<td>Accumulate customer information</td>
<td>Customer database</td>
<td>Name, address, phone, and e-mail for all customers; online customer registration</td>
</tr>
<tr>
<td>Provide after-sale customer support</td>
<td>Sales database</td>
<td>Customer ID, product, date, payment, shipment date</td>
</tr>
<tr>
<td>Coordinate marketing/advertising program</td>
<td>Ad server, e-mail server, e-mail campaign manager, ad banner manager</td>
<td>Site behavior log of prospects and customers linked to e-mail and banner ad campaigns</td>
</tr>
<tr>
<td>Understand marketing effectiveness</td>
<td>Site tracking and reporting system</td>
<td>Number of unique visitors, pages visited, products purchased, identified by marketing campaign</td>
</tr>
<tr>
<td>Provide production and supplier links</td>
<td>Inventory management system</td>
<td>Product and inventory levels, supplier ID and contact, order quantity data by product</td>
</tr>
</tbody>
</table>
from those of an ordinary retail store. The real difference lies in the system functionalities and information requirements: In an e-commerce site, the business objectives must be provided entirely in digital form without buildings or salespeople, twenty-four hours a day, seven days a week.

**SYSTEM DESIGN: HARDWARE AND SOFTWARE PLATFORMS**

Once you have identified the business objectives and system functionalities, and have developed a list of precise information requirements (see Table 4.1), you can begin to consider just how all this functionality will be delivered. You must come up with a **system design specification**—a description of the main components in a system and their relationship to one another. The system design itself can be broken down into two components: a logical design and a physical design. A **logical design** includes a data flow diagram that describes the flow of information at your e-commerce site, the processing functions that must be performed, and the databases that will be used. The logical design also includes a description of the security and emergency backup procedures that will be instituted, and the controls that will be used in the system.

A **physical design** translates the logical design into physical components. For instance, the physical design details the specific model of server to be purchased, the software to be used, the size of the telecommunications link that will be required, the way the system will be backed up and protected from outsiders, and so on.

Figure 4.3(a) presents a data flow diagram for a simple high-level logical design for a very basic Web site that delivers catalog pages in HTML in response to HTTP requests from the client’s browser, while Figure 4.3(b) shows the corresponding physical design. Each of the main processes can be broken down into lower level designs that are much more precise in identifying exactly how the information flows and what equipment is involved.

**BUILDING THE SYSTEM: IN-HOUSE VERSUS OUTSOURCING**

Now that you have a clear idea of both the logical and physical design for your site, you can begin considering how to actually build the site. There are many choices here. They range from outsourcing everything (including the actual systems analysis and design) to building everything in-house. **Outsourcing** means that you will hire an outside vendor to provide the services involved in building the site that you cannot perform with in-house personnel. Read *Insight on Business: The Small Entrepreneur’s Edge* for an example of one firm that chose this route. You also have a second decision to make: Will you host (operate) the site on your firm’s own servers or will you outsource the hosting to a Web host provider? These decisions are independent of each other, but they are usually considered at the same time. There are some vendors who will design, build, and host your site, while others will either build or host (but not both). Figure 4.4 illustrates the alternatives.
CHAPTER 4 Building an E-commerce Web Site

FIGURE 4.3 A LOGICAL AND PHYSICAL DESIGN FOR A SIMPLE WEB SITE

(a) Simple Data Flow Diagram
This data flow diagram describes the flow of information requests and responses for a simple Web site.

(b) Simple Physical Design
A physical design describes the hardware and software needed to realize the logical design.
Build Your Own versus Outsourcing

Let’s take the building decision first. If you elect to build your own site, you will need a multi-skilled staff of programmers, graphic artists, Web designers, and managers. You will also have to select and purchase hardware and software tools. There is a broad variety of tools available for building your own e-commerce site, ranging from those that help you build everything yourself from scratch, such as Dreamweaver and Front Page, to top-of-the-line prepackaged site-building tools that can create sophisticated sites customized to your needs, to prebuilt templates that merely require you to input text, graphics, and other data (see Figure 4.5). We will look more closely at the variety of e-commerce software available in Section 4.2.

The decision to build a Web site on your own has a number of risks. Given the complexity of features such as shopping carts, credit card authentication and processing, inventory management, and order processing, the costs involved are high, as are the risks of doing a poor job. You will be reinventing what other specialized firms have already built, and your staff may face a long, difficult learning curve, delaying your entry to market. Your efforts could fail (Albrecht and Gaffney, 1983). On the positive side, you may be better able to build a site that does exactly what you want, and more important, develop the in-house knowledge to allow you to change the site rapidly if necessary due to a changing business environment.

If you choose more expensive site-building packages, you will be purchasing state-of-the-art software that is well tested. You could get to market sooner. However, in order to make a sound decision, you will have to evaluate many different packages...
CHAPTER 4 Building an E-commerce Web Site

INSIGHT ON BUSINESS

On any given day, there are over 175,000 vinyl LPs on sale at eBay.com. How is a small two-person online record shop supposed to compete with thousands of other entrepreneurs using eBay? John Turton and his wife Marianne know it’s not easy—but it is possible. They operate Audiophileusa.com. Last year, they grossed $500,000 selling rare vinyl long-playing records to their 2,000 regular customers. Not bad for a business that runs out of a house and uses a 1,700-square-foot basement to store the inventory of over 30,000 albums and 1,500 compact discs, two-thirds of which have not yet been entered into the company database.

One factor in Audiophileusa.com’s success is personal attention to the needs of customers, and a very personalized approach to the service. Turton enters each album into his database himself. He refuses to hire a data entry clerk because he examines each record and CD for quality, imperfections, and what may be special notations on the CD, indicating the recording engineer or studio or other markings that add value and price. For every record on the Audiophileusa.com site, Turton either knows something about the music or will take time to write about the music for the site. Turton charges more than the local record shops for this service, but he feels he can give accurate market prices better than anyone on the Web because he spends so much time shopping for albums to add to his collection. He spends several days a week shopping in San Francisco Bay Area record stores, usually picking up 50–60 albums and spending $1,000 each week.

Sales plummeted after the World Trade Center attack, and 2002 sales did not grow. Because of slow sales, Turton decided to rebuild his Web site. Originally designed in 1994, it had become just a long list of albums, a “straggling mess” according to Turton. Turton and his wife shopped around for a Web site design firm, rejecting several bids of $25–$30,000, and instead chose Webscape Internet Management Ltd. in Vancouver, British Columbia. They spent $10,000 for a complete re-design. The new site is a dramatic change. The site features nine albums with pictures, a searchable database, and a virtual shopping cart. So far, customers are enthusiastic, but they want more frequent updates, a larger database, and more package deals or coupons. This calls for more re-design, a job that never seems to end for Web entrepreneurs.


and this can take a long time. You may have to modify the packages to fit your business needs and perhaps hire additional outside vendors to do the modifications. Costs rise rapidly as modifications mount (see Figure 4.6). If you choose the template route, you will be limited to the functionality already built into the templates, and you will not be able to add to the functionality or change it.
While sophisticated site development packages appear to reduce costs and increase speed to market, the modifications required to fit the package to your business needs rise exponentially. A $4,000 package can easily become a $40,000 to $60,000 development project.

Brick-and-mortar retailers in need of an e-commerce site generally design the site themselves (because they already have the skilled staff in place and have extensive investments in information technology capital such as databases and telecommunications), but they use outside vendors and consultants to build the sophisticated e-commerce applications. Small startups may build their own sites from scratch using in-house technical personnel in an effort to keep costs low. Medium-size startups will often purchase a sophisticated package and then modify it to suit their needs. Very small mom-and-pop firms seeking simple storefronts will use templates.

Host Your Own versus Outsourcing

Now let’s look at the hosting decision. Most businesses choose to outsource hosting and pay a company to host their Web site, which means that the hosting company is responsible for ensuring the site is “live,” or accessible, twenty-four hours a day. By agreeing to a monthly fee, the business need not concern itself with many of the technical aspects of setting up a Web server and maintaining it, nor with staffing needs.

You can also choose to co-locate. With a co-location agreement, your firm purchases or leases a Web server (and has total control over its operation), but locates the server in a vendor’s physical facility. The vendor maintains the facility, communications lines, and the machinery. See Table 4.2 for a list of some of the major hosting/co-location providers.

Hosting and co-location have become a commodity and a utility: Costs are driven by very large providers (such as IBM and Qwest) who can achieve large economies of scale by establishing huge “server farms” located strategically around the country and the globe. The capital expenditures for building these server farms, associated telecommunications links, and emergency power supplies are very large. Small ISPs also can be used as hosts, but service reliability is an issue. Will the small ISP be able to provide uninterrupted service, 24x7x365? Will they have service staff available when you need it?

The disadvantage of outsourcing hosting is that as the online business grows, the company may need more power and services than the hosting company can provide. That is why some corporations elect to do their own hosting. When you host your own site, you are in total control of the operation. Keep in mind that your costs may

<table>
<thead>
<tr>
<th>TABLE 4.2</th>
<th>KEY PLAYERS: HOSTING/CO-LOCATION SERVICES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cable and Wireless</td>
<td>Qwest Communications</td>
</tr>
<tr>
<td>Equinex</td>
<td>NTT/Verio</td>
</tr>
<tr>
<td>IBM Global Services</td>
<td>Rackspace</td>
</tr>
</tbody>
</table>
be higher than if you had used a large outsourcing firm. You will have to purchase hardware and software, have a physical facility, lease communications lines, hire a staff, and build security and backup capabilities.

TESTING THE SYSTEM

Once the system has been built and programmed, you will have to engage in a testing process. Depending on the size of the system, this could be fairly difficult and lengthy. Testing is required whether the system is outsourced or built in-house. A complex e-commerce site can have thousands of pathways through the site, each of which must be documented and then tested. Unit testing involves testing the site’s program modules one at a time. System testing involves testing the site as a whole, in the same way a typical user would when using the site. Because there is no truly “typical” user, system testing requires that every conceivable path be tested. Final acceptance testing requires that the firm’s key personnel and managers in marketing, production, sales, and general management actually use the system as installed on a test Internet or intranet server. This acceptance test verifies that the business objectives of the system as originally conceived are in fact working. It is important to note that testing is generally underbudgeted. As much as 50% of the software effort can be consumed by testing and rebuilding (usually depending on the quality of initial design).

IMPLEMENTATION AND MAINTENANCE

Most people unfamiliar with systems erroneously think that once an information system is installed, the process is over. In fact, while the beginning of the process is over, the operational life of a system is just beginning. Systems break down for a variety of reasons—most of them unpredictable. Therefore, they need continual checking, testing, and repair. Systems maintenance is vital, but sometimes not budgeted for. In general, the annual system maintenance cost will roughly parallel the development cost. A $40,000 e-commerce site will likely require a $40,000 annual expenditure to maintain. Very large e-commerce sites experience some economies of scale, so that, for example, a $1 million site will likely require a maintenance budget of $500,000 to $700,000.

Why does it cost so much to maintain an e-commerce site? Unlike payroll systems, for example, e-commerce sites are always in a process of change, improvement, and correction. Studies of traditional systems maintenance have found 20% of the time is devoted to debugging code and responding to emergency situations (a new server was installed by your ISP, and all your hypertext links were lost and CGI scripts disabled—the site is down!) (Lientz and Swanson, 1980; Banker and Kemerer, 1989). Another 20% of the time is concerned with changes in reports, data files, and links to backend databases. The remaining 60% of maintenance time is devoted to general
administration (making product and price changes in the catalog) and making changes and enhancements to the system. E-commerce sites are never finished: They are always in the process of being built and rebuilt. They are dynamic—much more so than payroll systems.

The long-term success of an e-commerce site will depend on a dedicated team of employees (the Web team) whose sole job is to monitor and adapt the site to changing market conditions. The Web team must be multi-skilled; it will typically include programmers, designers, and business managers drawn from marketing, production, and sales support. One of the first tasks of the Web team is to listen to customers' feedback on the site and respond to that feedback as necessary. A second task is to develop a systematic monitoring and testing plan to be followed weekly to ensure all the links are operating, prices are correct, and pages are updated. A large business may have thousands of Web pages, many of them interlinked, that require systematic monitoring. Other important tasks of the Web team include benchmarking (a process in which the site is compared with those of competitors in terms of response speed, quality of layout, and design) and keeping the site current on pricing and promotions. The Web is a competitive environment where you can very rapidly frustrate and lose customers with a dysfunctional site (see Insight on Technology: Buying Something You Can’t See).

**FACTORS IN OPTIMIZING WEB SITE PERFORMANCE**

The purpose of a Web site is to deliver content to customers and to complete transactions. The faster and more reliably these two objectives are met, the more effective the Web site is from a commerce perspective. The optimization of Web site performance is more complicated than it seems and involves at least three factors: page content, page generation, and page delivery (see Figure 4.7). In this chapter, we describe the software and hardware choices you will need to make in building an e-commerce site; these are also important factors in Web site optimization.

Using efficient styles and techniques for page design and content can reduce response times by 2–5 seconds. Simple steps include reducing unnecessary HTML comments and white space, using more efficient graphics, and avoiding unnecessary links to other pages in the site. Page generation speed can be enhanced by segregating computer servers to perform dedicated functions (such as static page generation, application logic, media servers, database servers), and using various devices from vendors to speed up these servers. Using a single server or multiple servers to perform multiple tasks reduces throughput by over 50%. Page delivery can be speeded up by using edge caching services such as Akamai, or specialized content delivery networks such as RealNetworks, or by increasing local bandwidth. We will discuss some of these factors throughout the chapter, but a full discussion of Web site optimization is beyond the scope of this text.
BUYING SOMETHING YOU CAN’T SEE

When you buy flowers online, you can usually see a picture of the flower basket you ordered, and when you shop for a car online, you can see a picture of the car, plus lengthy descriptions of options and capabilities. But insurance is a different product altogether. Insurance is an intangible product with literally thousands of options within each of the major product categories (auto, house, rental, casualty, business, and health). As financial service firms move online, they have demonstrated that banking, investing, and advising services can be brought to the online consumer. However, insurance illustrates a class of products that are difficult to describe or compare and therefore difficult to sell online.

InsWeb.com, founded in 1995, provides consumers with an online marketplace where they can shop for, compare, and purchase health, auto, renters, and other insurance products. In 2002, InsWeb had revenues of $25 million, and a net loss of $4.5 million, a vast improvement over a loss of $45 million in 2001. InsWeb derives commission and referral revenue from 36 auto insurers and 20 health insurers. Deals can be made directly online, but many transactions occur offline in direct negotiations with the consumer and the insurance carrier.

At the Insweb site, the difficulties of purchasing insurance online are apparent. For a simple auto insurance policy with one driver and one car, policies can be compared and quoted. But if you have multiple drivers, several teenage children, or multiple vehicles, you likely want to discuss the options with an agent before making a choice and purchase. In health insurance, a similar situation prevails. Over twenty policies are listed, but to compare them you need to click on each one, read extensive documentation, and spend a lot of time trying to understand what you have just read before calling an agent.

A Learning Center provides articles about different kinds of insurance, but the articles are generic and are not necessarily helpful in guiding you to the right choice. What’s needed is an onsite expert who rates policies, or a system of rating. The site might imitate Amazon’s practice of showing what other consumers like yourself are buying, or provide consumer reviews of policies and most important how well the insurance companies service their accounts.

InsWeb demonstrates the challenges of designing a Web site to fit the product being sold, or alternatively, the need to redesign the product so it can be sold online. Yet a third alternative might be to develop intelligent software that would guide the client to the right choice with plenty of direct consumer experience feedback so that he or she could trust the software.

WEB SITE BUDGETS

While how much you spend to build a Web site depends on how much you can afford, and, of course, the size of the opportunity, Figure 4.8 provides some idea of the relative size of various Web site costs according to a survey of 125 Web site managers. About 75% of the costs of Web sites involve technology costs—development, software licenses, and hardware. About 18% of costs will be for design and development, and 6% for marketing the site.

4.2 CHOOSING SERVER SOFTWARE

What you are able to do at an e-commerce site is largely a function of the software. As a business manager in charge of building the site, you will need to know some basic information about e-commerce software. The more sophisticated the software and the more ways you can sell goods and services, the more effective your business will be. In this section, we will describe the software needed to operate a contemporary e-commerce site. Then, in Section 4.3, we discuss the hardware you will need to handle the demands of the software.
SIMPLE VERSUS MULTI-TIERED WEB SITE ARCHITECTURE

Prior to the development of e-commerce, Web sites simply delivered Web pages to users who were making requests through their browsers for HTML pages. Web site software was appropriately quite simple—it consisted of a server computer running basic Web server software. We might call this arrangement a single-tier system architecture. **System architecture** refers to the arrangement of software, machinery, and tasks in an information system needed to achieve a specific functionality (much like a home’s architecture refers to the arrangement of building materials to achieve a particular functionality).

However, the development of e-commerce required a great deal more functionality, such as the ability to respond to user input (name and address forms), take customer orders for goods and services, clear credit card transactions on the fly, consult price and product databases, and even adjust advertising on the screen based on user characteristics. This kind of extended functionality required the development of **Web application servers** and a multi-tiered system architecture to handle the processing loads. Web application servers, described more fully later in this section, are specialized software programs that perform a wide variety of transaction processing required by e-commerce.

In addition to having specialized application servers, e-commerce sites must be able to pull information from and add information to pre-existing corporate databases. These older databases that predate the e-commerce era are called **backend** or **legacy** databases. Corporations have made massive investments in these systems to store...
two-tier architecture
e-commerce system architecture in which a Web server responds to requests for Web pages and a database server provides backend data storage.

multi-tier architecture
e-commerce system architecture in which the Web server is linked to a middle-tier layer that typically includes a series of application servers that perform specific tasks as well as to a backend layer of existing corporate systems.

their information on customers, products, employees, and vendors. These backend systems constitute an additional layer in a multi-tiered site.

Figure 4.9 illustrates a simple two-tier and more complex multi-tier e-commerce system architecture. In two-tier architecture, a Web server responds to requests for Web pages and a database server provides backend data storage. In a multi-tier architecture, in contrast, the Web server is linked to a middle-tier layer that typically
includes a series of application servers that perform specific tasks, as well as to a backend layer of existing corporate systems containing product, customer, and pricing information. A multi-tiered site typically employs several or more physical computers, each running some of the software applications and sharing the work load across many physical computers.

In the remainder of this section, we will describe basic Web server software functionality and the various types of Web application servers.

WEB SERVER SOFTWARE

All e-commerce sites require basic Web server software to answer requests from customers for HTML and XML pages. The leading Web server software choices are shown in Figure 4.10.

When you choose Web server software, you will also be choosing an operating system for your site’s computers. The leading Web server software, with over 60% of the market, is Apache, which works only with the UNIX operating system. UNIX is the original programming language of the Internet and Web. Apache was developed by a worldwide community of Internet innovators. Apache is free and can be downloaded from many sites on the Web; it also comes installed on most IBM Web servers. Literally thousands of programmers have worked on Apache over the years; thus, it is extremely stable. There are thousands of utility software programs written for Apache that can provide all the functionality required for a contemporary e-commerce site. In order to use Apache, you will need staff who are knowledgeable in the operation of the UNIX operating system (or its PC variant, Linux).

![Figure 4.10: Key Players in Web Server Software](image)

This diagram illustrates the relative market share of the most popular Web server software. As you can see, Apache HTTP dominates the market.

Microsoft Internet Information Server (IIS) is the second major Web server software available, albeit a distant second to Apache, with about 27% of the market. IIS is based on the Windows 2000 operating system and is compatible with a wide selection of Microsoft utility and support programs.

There are also at least one hundred other smaller providers of Web server software, most of them based on UNIX. Note that the choice of Web server has little effect on users of your system. The pages they see will look the same regardless of the development environment. There are many advantages to the Microsoft suite of development tools—they are integrated, powerful, and easy to use. The UNIX operating system, on the other hand, is exceptionally reliable and stable, and there is a worldwide open software community that develops and tests UNIX-based Web server software.

Table 4.3 shows the basic functionality provided by all Web servers.

### Site Management Tools

In Chapter 3, we described most of the basic functionality of the Web servers listed in Table 4.3. One functionality not described previously is site management tools. **Site management tools** are essential if you want to keep your site working, and if you want to understand how well it is working. Site management tools verify that links on pages are still valid and also identify **orphan files**, or files on the site that are not linked to any pages. By surveying the links on a Web site, a site management tool can quickly report on potential problems and errors that users may encounter. Links to URLs that have moved or been deleted are called **dead links**; these can cause error

<table>
<thead>
<tr>
<th>TABLE 4.3</th>
<th>BASIC FUNCTIONALITY PROVIDED BY WEB SERVERS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FUNCTIONALITY</strong></td>
<td><strong>DESCRIPTION</strong></td>
</tr>
<tr>
<td>Processing of HTTP requests</td>
<td>Receive and respond to client requests for HTML pages</td>
</tr>
<tr>
<td>Security services (Secure Sockets Layer)</td>
<td>Verify username and password; process certificates and private/public key information required for credit card processing and other secure information</td>
</tr>
<tr>
<td>File Transfer Protocol</td>
<td>Permits transfer of very large files from server to server</td>
</tr>
<tr>
<td>Search engine</td>
<td>Indexing of site content; keyword search capability</td>
</tr>
<tr>
<td>Data capture</td>
<td>Log file of all visits, time, duration, and referral source</td>
</tr>
<tr>
<td>E-mail</td>
<td>Ability to send, receive, and store e-mail messages</td>
</tr>
<tr>
<td>Site management tools</td>
<td>Calculate and display key site statistics, such as unique visitors, page requests, and origin of requests; check links on pages</td>
</tr>
</tbody>
</table>
messages for users trying to access that link. Regularly checking that all links on a site are operational helps prevent irritation and frustration in users who may decide to take their business elsewhere to a better functioning site.

Additional site management software and services, such as those provided by Webtrends.com, can be purchased in order to more effectively monitor customer purchases and marketing campaign effectiveness, as well as keep track of standard hit counts and page visit information. Page 216 contains several screenshots that illustrate the different types of functionality provided by WebTrends software.

**Dynamic Page Generation Tools**

One of the most important innovations in Web site operation has been the development of dynamic page generation tools. Prior to the development of e-commerce, Web sites primarily delivered unchanging static content in the form of HTML pages. While this capability might be sufficient to display pictures of products, consider all the elements of a typical e-commerce site today by reviewing Table 4.1, or visit what you believe is an excellent e-commerce site. The content of successful e-commerce sites is always changing, often day by day. There are new products and promotions, changing prices, news events, and stories of successful users. E-commerce sites must intensively interact with users, who not only request pages, but also request product, price, availability, and inventory information. One of the most dynamic sites is eBay.com—the auction site. There, the content is changing minute by minute. E-commerce sites are just like real markets—they are dynamic.

The dynamic and complex nature of e-commerce sites requires a number of specialized software applications in addition to static HTML pages. Perhaps one of the most important is dynamic page generation software. With dynamic page generation, the contents of a Web page are stored as objects in a database, rather than being hard-coded in HTML. When the user requests a Web page, the contents for that page are then fetched from the database. The objects are retrieved from the database using CGI (Common Gateway Interface), ASP (Active Server Pages), JSP (Java Server Pages), or other server-side programs. CGI, ASP, and JSP are described in the last section of this chapter. This technique is much more efficient than working directly in HTML code. It is much easier to change the contents of a database than it is to change the coding of an HTML page. A standard data access method called Open Database Connectivity (ODBC) makes it possible to access any data from any application regardless of what database is used. ODBC is supported by most of the large database suppliers such as Oracle, Sybase, and IBM. ODBC makes it possible for HTML pages to be linked to backend corporate databases regardless of who manufactured the database. Web sites must be able to pull information from, and add information to, these databases. For example, when a customer clicks on a picture of a pair of boots, the site can access the product catalog database stored in a DB2 database, and access the inventory database stored in an Oracle database to confirm that the boots are still in stock and to report the current price (Hughes and Birznieks, 2001).
Using a sophisticated site management program such as WebTrends, managers can quickly understand the return on investment in specific online campaigns, the interests of different types of customers, the lifetime value of customers produced by different online sources, and details of the purchase process, from viewing charts to checking out.
Dynamic page generation gives e-commerce several significant capabilities that generate cost and profitability advantages over traditional commerce. Dynamic page generation lowers *menu costs* (the costs incurred by merchants for changing product descriptions and prices). Dynamic page generation also permits easy online *market segmentation*—the ability to sell the same product to different markets. The same capability makes possible nearly *cost-free price discrimination*—the ability to sell the same product to different customers at different prices. For instance, you might want to sell the same product to corporations and government agencies—but use different marketing themes. Based on a cookie you placed on client files, or in response to a question on your site that asks visitors if they are from a government agency or a corporation, you would be able to use different marketing and promotional materials for corporate clients and government clients. You might want to reward loyal customers with lower prices, say on CDs or musical tracks, and charge full price to first-time buyers. In summary, dynamic page generation allows you to approach different customers with different messages and prices.

**APPLICATION SERVERS**

**Web application servers** are software programs that provide the specific business functionality required of a Web site. The basic idea of application servers is to isolate the business applications from the details of displaying Web pages to users on the front end and the details of connecting to databases on the back end. Application servers are a kind of *middleware software* that provides the glue connecting traditional corporate systems to the customer as well as all the functionality needed to conduct e-commerce. In the early years, a number of software firms developed specific separate programs for each function, but increasingly, these specific programs are being replaced by integrated software tools that combine all the needed functionality for an e-commerce site into a single development environment, a packaged software approach.

Table 4.4 illustrates the wide variety of application servers available in the marketplace. The table focuses on “sell side” servers that are designed to enable selling products on the Web. So-called “buy side” and “link” servers focus on the needs of businesses to connect with partners in their supply chains or find suppliers for specific parts and assemblies. These buy side and link servers are described more fully in Chapter 12, *B2B E-commerce, Supply Chain Management, and Collaborative Commerce*.

There are several thousand software vendors that provide application server software. For UNIX environments, many of these capabilities are available free on the Internet from various sites. Most businesses—faced with this bewildering array of choices—choose to use integrated software tools called *merchant server software*.

**E-COMMERCE MERCHANT SERVER SOFTWARE FUNCTIONALITY**

E-commerce merchant server software provides the basic functionality needed for online sales, including an online catalog, order taking via an online shopping cart, and online credit card processing.
CHAPTER 4  BUILDING AN E-COMMERCE WEB SITE

Online Catalog

A company that wants to sell products on the Web must have a list, or **online catalog**, of its products, available on its Web site. Merchant server software typically includes a database capability that will allow for construction of a customized online catalog. The complexity and sophistication of the catalog will vary depending on the size of the company and its product lines. Small companies, or companies with small product lines, may post a simple list with text descriptions and perhaps color photos. A larger site might decide to add sound, animations, or videos (useful for product demonstrations) to the catalog, or interactivity, such as customer service representatives available via Instant Messaging to answer questions.

Shopping Carts

Online **shopping carts** are much like their real-world equivalent; both allow shoppers to set aside desired purchases in preparation for checkout. The difference is that

<table>
<thead>
<tr>
<th>APPLICATION SERVER</th>
<th>FUNCTIONALITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Catalog display</td>
<td>Provides a database for product descriptions and prices</td>
</tr>
<tr>
<td>Transaction processing (shopping cart)</td>
<td>Accepts orders and clears payments</td>
</tr>
<tr>
<td>List server</td>
<td>Creates and serves mailing lists and manages e-mail marketing campaigns</td>
</tr>
<tr>
<td>Proxy server</td>
<td>Monitors and controls access to main Web server; implements firewall protection</td>
</tr>
<tr>
<td>Mail server</td>
<td>Manages Internet e-mail</td>
</tr>
<tr>
<td>Audio/video server</td>
<td>Stores and delivers streaming media content</td>
</tr>
<tr>
<td>Chat server</td>
<td>Creates an environment for online real-time text and audio interactions with customers</td>
</tr>
<tr>
<td>News server</td>
<td>Provides connectivity and displays Internet news feeds</td>
</tr>
<tr>
<td>Fax server</td>
<td>Provides fax reception and sending using a Web server</td>
</tr>
<tr>
<td>Groupware server</td>
<td>Creates workgroup environments for online collaboration</td>
</tr>
<tr>
<td>Database server</td>
<td>Stores customer, product, and price information</td>
</tr>
<tr>
<td>Ad server</td>
<td>Maintains Web-enabled database of advertising banners that permits customized and personalized display of advertisements based on consumer behavior and characteristics</td>
</tr>
<tr>
<td>Auction server</td>
<td>Provides a transaction environment for conducting online auctions</td>
</tr>
<tr>
<td>B2B Server</td>
<td>Implements buy, sell, and link marketplaces for commercial transactions</td>
</tr>
</tbody>
</table>

**TABLE 4.4 APPLICATION SERVERS AND THEIR FUNCTION**
Choosing Server Software

the online variety is part of a merchant server software program residing on the Web server, and allows consumers to select merchandise, review what they have selected, edit their selections as necessary, and then actually make the purchase by clicking a button. Shopping cart data is automatically stored by the merchant server software.

Credit Card Processing

A site’s shopping cart typically works in conjunction with credit card processing software, which verifies the shopper’s credit card and then puts through the debit to the card and the credit to the company’s account at checkout. Integrated e-commerce software suites typically supply the software for this function. Otherwise, you will have to make arrangements with a variety of credit card processing banks and intermediaries.

MERCHANT SERVER SOFTWARE PACKAGES (E-COMMERCE SUITES)

Rather than build your site from a collection of disparate software applications, it is easier, faster, and generally more cost effective to purchase a merchant server software package (also called an e-commerce server suite). Merchant server software/e-commerce suites offer an integrated environment that promises to provide most or all of the functionality and capabilities you will need to develop a sophisticated, customer-centric site. E-commerce suites come in three general ranges of price and functionality.

Basic packages for elementary e-commerce business applications are provided by Bizland, Hypermart, and Yahoo Stores.

Midrange suites include IBM’s WebSphere Commerce Professional Entry Edition and Microsoft’s Commerce Server 2002. High-end enterprise solutions for large global firms are provided by IBM Websphere’s Commerce Professional Edition, Broadvision One-to-One Commerce, Interworld’s Commerce Exchange 6.0 and others. There are over a hundred software firms that provide e-commerce suites. Table 4.5 lists some of the most widely adopted midrange and high-end e-commerce suites.

Choosing an E-commerce Suite

With all of these vendors, how do you choose the right one? Evaluating these tools and making a choice is one of the most important and uncertain decisions you will make in building an e-commerce site. The real costs are hidden—they involve training your staff to use the tools and integrating the tools into your business processes and organizational culture (Glass, 1999; Valdes, 2000). The following are some of the key factors to consider:

• Functionality
• Support for different business models
• Business process modeling tools
• Visual site management tools and reporting
For instance, although e-commerce suites promise to do everything, your business may require special *functionality*—such as streaming audio and video. You will need a list of business functionality requirements. Your business may involve several different business models—such as a retail side and a business-to-business side; you may run auctions for stock excess as well as fixed-price selling. Be sure the package can *support all of your business models*. You may wish to change your business processes—such as order taking and order fulfillment. Does the suite contain *tools for modeling business process and work flows*? Understanding how your site works will require *visual reporting tools* that make its operation transparent to many different people in your business. A poorly designed software package will drop off significantly in performance as visitors and transactions expand into the thousands per hour, or minute. Check for *performance and scalability* by stress testing a pilot edition or obtaining data from the vendor about performance under load. You will have to connect the e-commerce suite to your traditional business systems. How will this *connection to existing systems* be made, and is your staff skilled in making the connection? Because of the changing technical environment—in particular, changes in mobile commerce platforms—it is important to document exactly *what standards are supported* by the suite now, and what is the migration path toward the future. Finally,
Choosing the Hardware for an E-commerce Site

Your e-commerce site may have to work both globally and locally. You may need a foreign language edition using foreign currency denominations. And you will have to collect sales taxes across many local, regional, and national tax systems. Does the e-commerce suite support this level of globalization and localization?

**Building Your Own E-commerce Software**

If you decide to attempt to build your own e-commerce capabilities, there are a number of increasingly powerful tools available to help you. For very simple e-commerce sites, services such as Freemerchant.com offer a free turnkey (complete) solution for building a simple online store. BigStep.com takes users step by step through the process of building an online store. eCongo.com, Tripod.com, and Yahoo Store all provide easy-to-use site-building tools and e-commerce templates for simple e-commerce sites. An e-commerce template is a predesigned Web site that allows users to customize the look and feel of the site to fit their business needs and provides a standard set of functionality. Most templates today contain ready-to-go site designs with built-in commerce features.

There are many providers of sophisticated (and much more expensive) e-commerce site development tools. For example, BEA Systems bundles basic e-commerce page and site creation tools with a number of powerful e-commerce applications such as customer and product databases, onsite customer tracking, and customer relationship management tools; it also provides interfaces with backend databases containing product and price information.

Of course, you can build everything for your site from scratch yourself, but this requires that you have the necessary personal knowledge or support staff. You can use Microsoft FrontPage or Macromedia's Dreamweaver to build your basic pages for the catalog. Then, in order to take orders and interact with the client, you will need to know how to write CGI scripts, and you will have to understand how to work with a database package such as SQL. Often, the challenges of building an e-commerce site are so great that businesses decide to use e-commerce suites instead. Once you have selected the software tools needed for your e-commerce site, you will have to select the computer hardware and telecommunications links needed to realize your vision of the site.

4.3 **Choosing the Hardware for an E-commerce Site**

As the manager in charge of building an e-commerce site, you will be held accountable for its performance. Whether you host your own site or outsource the hosting
and operation of your site, you will need to understand certain aspects of the computing hardware platform. The hardware platform refers to all the underlying computing equipment that the system uses to achieve its e-commerce functionality. Your objective is to have enough platform capacity to meet peak demand (avoiding an overload condition), but not so much platform that you are wasting money. Failing to meet peak demand can mean your site is slow, or actually crashes. Remember, the Web site may be your only or principal source of cash flow. How much computing and telecommunications capacity is enough to meet peak demand? How many hits per day can your site sustain?

To answer these questions, you will need to understand the various factors that affect the speed, capacity, and scalability of an e-commerce site.

**RIGHT-SIZING YOUR HARDWARE PLATFORM: THE DEMAND SIDE**

The most important factor affecting the speed of your site is the demand that customers put on the site. Table 4.6 lists the most important factors to consider when estimating the demand on a site.

Demand on a Web site is fairly complex and depends primarily on the type of site you are operating. The number of simultaneous users in peak periods, the nature of customer requests, the type of content, the required security, the number of items in inventory, the number of page requests, and the speed of legacy applications that may be needed to supply data to the Web pages are all important factors in overall demand on a Web site system.

Certainly, one important factor to consider is the number of simultaneous users who will likely visit your site. In general, the load created by an individual customer on a server is typically quite limited and short-lived. A Web session initiated by the typical user is stateless, meaning that the server does not have to maintain an ongoing, dedicated interaction with the client. A Web session typically begins with a page request, then a server replies, and the session is ended. The sessions may last from tenths of a second per user, to a minute. Nevertheless, system performance does degrade as more and more simultaneous users request service. Fortunately, degradation (measured as “transactions per second” and “latency” or delay in response) is fairly graceful over a wide range, up until a peak load is reached and service quality becomes unacceptable (see Figure 4.11).

In general, a robust, single-processor Web server (with, for example, a Pentium 4 at 2 GHz [gigahertz]), serving only static Web pages, can handle about 8,000 concurrent users. Serving up static Web pages is I/O intensive, which means it requires input/output operations rather than heavy-duty processing power. As a result, Web site performance is constrained primarily by the server’s input/output (I/O) limitations and the telecommunications connection, rather than speed of the processor.

We discuss some of the steps you can take to make sure that you stay within an acceptable service quality. One step is to simply purchase a server with faster CPU
TABLE 4.6  FACTORS IN RIGHT-SIZING AN E-COMMERCE PLATFORM

<table>
<thead>
<tr>
<th>SITE TYPE</th>
<th>PUBLISH/</th>
<th>SHOPPING</th>
<th>CUSTOMER</th>
<th>TRADING</th>
<th>WEB SERVICES/B2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Examples</td>
<td>WSJ.com</td>
<td>Amazon.com</td>
<td>NetBank.com</td>
<td>E-Trade.com</td>
<td>Ariba e-procurement exchanges</td>
</tr>
<tr>
<td>Content</td>
<td>Dynamic</td>
<td>Catalog</td>
<td>Data in legacy applications</td>
<td>Time sensitive</td>
<td>Data in legacy applications</td>
</tr>
<tr>
<td></td>
<td>Multiple authors</td>
<td>Dynamic items</td>
<td>Multiple data sources</td>
<td>High volatility</td>
<td>High volatility</td>
</tr>
<tr>
<td></td>
<td>High volume</td>
<td>User profiles with data mining</td>
<td>Multiple suppliers and consumers</td>
<td>Multiple data transactions</td>
<td>Complex transactions</td>
</tr>
<tr>
<td></td>
<td>Not user specific</td>
<td>Privacy</td>
<td>Complex transactions</td>
<td>Privacy</td>
<td>Privacy</td>
</tr>
<tr>
<td>Security</td>
<td>Low</td>
<td>Privacy</td>
<td>Privacy</td>
<td>Privacy</td>
<td>Privacy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Non-repudiation</td>
<td>Non-repudiation</td>
<td>Non-repudiation</td>
<td>Non-repudiation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Integrity</td>
<td>Integrity</td>
<td>Integrity</td>
<td>Integrity</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Authentication</td>
<td>Authentication</td>
<td>Authentication</td>
<td>Authentication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Regulations</td>
<td>Regulations</td>
<td>Regulations</td>
<td>Regulations</td>
</tr>
<tr>
<td>Percent Secure Pages</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>High</td>
<td>Medium</td>
</tr>
<tr>
<td>Cross Session Information</td>
<td>No</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Searches</td>
<td>Dynamic</td>
<td>Dynamic</td>
<td>Low volume</td>
<td>Moderate to high</td>
<td>Moderate to high</td>
</tr>
<tr>
<td></td>
<td>Low volume</td>
<td>High volume</td>
<td>Low volume</td>
<td>Moderate to high</td>
<td>Moderate to high</td>
</tr>
<tr>
<td>Unique items (SKUs)</td>
<td>High</td>
<td>Medium to high</td>
<td>Medium</td>
<td>High</td>
<td>Medium to high</td>
</tr>
<tr>
<td>Transaction Volume</td>
<td>Moderate</td>
<td>Moderate to high</td>
<td>Moderate</td>
<td>High to extremely high</td>
<td>Moderate</td>
</tr>
<tr>
<td>Legacy Integration Complexity</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>Page views (hits)</td>
<td>High to very high</td>
<td>Moderate to high</td>
<td>Moderate to low</td>
<td>Moderate to high</td>
<td>Moderate</td>
</tr>
</tbody>
</table>

processors, multiple CPU processors, or larger hard disk drives. Windows 2000 Server, for instance, can support up to eight processors. However, the improvement that results is not linear and at some point becomes cost ineffective. Figure 4.12 shows the performance of a Windows 2000 Server as processors are added from a single processor up to eight processors. By increasing processors by a factor of eight, you get only three times more load capacity.

A second factor to consider on the demand side is the user profile, which refers to the nature of customer requests and customer behavior on your site (how many pages customers request and the kind of service they want). The 2003 UCLA Internet
As the number of concurrent users (N) rises, the transaction rate (T) rises linearly until an inflection point (X) is reached, after which performance falls at a nonlinear rate until a crash is experienced. Likewise, latency increases to a point where it becomes exponential and service quality is unacceptable.

Report study found over 75% of visitors to the typical e-commerce site simply browse—requesting static Web pages (see Table 4.7). Web servers can be very efficient at serving static Web pages (over three-fourths of the load as noted in Table 4.7). However, as customers request more advanced services, such as searches of site, registration, order taking via shopping carts, or downloads of large multimedia audio and video files, all of which require more processing power, performance can deteriorate rapidly.

The nature of the content your site offers is a third factor to consider. If your site uses dynamic page generation, then the load on the processor rises rapidly and performance will degrade. Dynamic page generation and business logic (such as a shopping cart) are CPU-intensive operations—they require a great deal of processing power. For instance, a site with only dynamic page content can expect performance of a single processor server to fall to one-tenth the levels described in Figure 4.12. Instead of effectively serving 8,000 users, you can only service 1,000 concurrent users. Any interaction with the user requiring access to a database—filling out forms, adding to carts, purchasing, and questionnaires—puts a heavy processing load on the server.

Figure 4.13 shows the impact of increasing dynamic page content from 25% to 50% on a variety of processor configurations.
The typical single-processor Web static page server can handle about 8,000 concurrent users. With eight processors, the same computer could handle about 25,000 concurrent users.

A final factor to consider is the telecommunications link that your site has to the Web, and also the changing nature of the client connection to the Web. Figure 4.14 shows that the number of hits per second your site can handle depends on the bandwidth connection between your server and the Web. The larger the bandwidth available, the more customers can simultaneously hit your site. For example, if your connection to the Web is a 1.5 Mbps DSL line, the maximum number of visitors per second is 30,000.

### Table 4.7 Visitor Profile at Typical E-Commerce Sites

<table>
<thead>
<tr>
<th>Visitor Activity</th>
<th>Percentage of Visitors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browse</td>
<td>76%</td>
</tr>
<tr>
<td>Search for content</td>
<td>51%</td>
</tr>
<tr>
<td>Shop and purchase goods</td>
<td>44%</td>
</tr>
</tbody>
</table>

second for 1 kilobyte files is probably about 100. Most businesses host their sites at an ISP or other provider that contractually is (or should be) obligated to provide enough bandwidth for your site to meet peak demands. However, there are no guarantees and ISPs can blame Web congestion for their own bandwidth limitations. Check your ISP’s bandwidth and your site performance daily.

While server bandwidth connections are less a constraint today with the widespread deployment of fiber-optic cable, the connection to the client is improving. At the end of 2002, there were approximately 17 million broadband households in the United States, and this number is expected to increase to 30 million by 2004 (eMarketer, 2003). This means they will be able to make far more frequent requests and demand far richer content and experiences from your site. This demand will translate quickly into dynamic content and the need for additional capacity.

**RIGHT-SIZING YOUR HARDWARE PLATFORM: THE SUPPLY SIDE**

Once you estimate the likely demand on your site, you will need to consider how to scale up your site to meet demand. **Scalability** refers to the ability of a site to increase in size as demand warrants. There are three steps you can take to meet the demands for service at your site: scale hardware vertically, scale hardware horizontally, and/or

---

**Figure 4.13: The Impact of Dynamic Page Content on Web Servers**

This figure illustrates the impact of dynamic page content on IBM RS6000 servers with different processors, ranging from a single processor to dual processor, quad processor, and finally to a parallel processor computer. Doubling dynamic content from 25% to 50% typically reduces throughput by up to one third.

improve the processing architecture of the site (see Table 4.8). Vertical scaling refers to increasing the processing power of individual components. Horizontal scaling refers to employing multiple computers to share the workload (and increase the “footprint” of the installation) (IBM, 2002).

You can scale your site vertically by upgrading the servers from a single processor to multiple processors (see Figure 4.15). For instance, you can change your hardware from a single-processor Pentium 4, to a dual or quad processor with two Pentium 4 processors. You can keep adding processors to a computer depending on the operating system and upgrade to faster chip speeds as well.

There are two drawbacks to vertical scaling. First, it can become expensive to purchase new computers with every growth cycle, and second, your entire site becomes dependent on a small number of very powerful computers. If you have two such computers and one goes down, half of your site, or perhaps your entire site, may become unavailable.

Horizontal scaling involves adding multiple single-processor servers to your site and balancing the load among the servers. You can also then partition the load so that
some servers handle only requests for HTML or ASP pages, while others are dedicated to handling database applications. You will need special load-balancing software (provided by a variety of vendors such as Cisco, Microsoft, and IBM) to direct incoming requests to various servers. (see Figure 4.16).

There are many advantages to horizontal scaling. It is inexpensive and often can be accomplished using older PCs that otherwise would be disposed of. Horizontal scaling also introduces redundancy—if one computer fails, chances are that another computer can pick up the load dynamically. However, when your site grows from a single computer to perhaps ten to twenty computers, the size of the physical facility required (the “footprint”) increases and there is added management complexity.

---

**TABLE 4.8**

**EIGHT VERTICAL AND HORIZONTAL SCALING TECHNIQUES**

<table>
<thead>
<tr>
<th>TECHNIQUE</th>
<th>APPLICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use a faster computer</td>
<td>Applies to edge servers, presentation servers, data servers, etc.</td>
</tr>
<tr>
<td>Create a cluster of computers</td>
<td>Use computers in parallel to balance loads</td>
</tr>
<tr>
<td>Use appliance servers</td>
<td>Special-purpose computers optimized for their task</td>
</tr>
<tr>
<td>Segment workload</td>
<td>Segment incoming work to specialized computers</td>
</tr>
<tr>
<td>Batch requests</td>
<td>Combine related requests for data into groups, process as a group</td>
</tr>
<tr>
<td>Manage connections</td>
<td>Reduce connections between processes and computers to a minimum</td>
</tr>
<tr>
<td>Aggregate user data</td>
<td>Aggregate user data from legacy applications in single data pools</td>
</tr>
<tr>
<td>Cache</td>
<td>Store frequently used data in cache rather than on the disk</td>
</tr>
</tbody>
</table>

---

You can scale a site vertically by both improving the processors and adding additional CPUs into a single physical server.
A third alternative—improving the processing architecture—is a combination of vertical and horizontal scaling, combined with artful design decisions. Table 4.9 lists some of the more common steps you can take to greatly improve performance of your site.

Most of these steps involve splitting the workload into I/O-intensive activities (such as serving Web pages), and CPU-intensive activities (such as taking orders).

---

**FIGURE 4.16** HORIZONTALLY SCALING A SYSTEM

You can horizontally scale a system to meet demands by adding inexpensive single-processor servers to the site and using load-balancing software to allocate incoming customer requests to the correct server, shown in the diagram as a Cisco LocalDirector.
Once you have this work separated, you can fine-tune the servers for each type of load. One of the least expensive fine-tuning steps is to simply add RAM to a few servers and store all your HTML pages in RAM. This reduces load on your hard drives and increases speed dramatically. RAM is thousands of times faster than hard disks, and RAM is inexpensive. The next most important step is to move your CPU-intensive activities, such as order taking, onto a high-end, multiple processor server that is totally dedicated to handling orders and accessing the necessary databases.

Taking these steps can permit you to reduce the number of servers required to service 10,000 concurrent users from 100 down to 20, according to one estimate.

### 4.4 OTHER E-COMMERCE SITE TOOLS

Now that you understand the key factors that affect the speed, capacity, and scalability of your site, we can consider some other important requirements for your Web site. You will need a coherent Web site design effort that makes business sense—not necessarily a site to wow visitors or excite them, but to sell them something. You will also need to know how to build active content and interactivity into your site—not just display static HTML pages. You will definitely want to be able to track customers who come, leave, and return to your site in order to be able to greet return visitors (“Hi Sarah, Glad to Have You Return!”). You will also want to track customers throughout your site so you can personalize and customize their experience. Finally, you will
need to establish a set of information policies for your site—privacy, accessibility, and access to information policies.

In order to achieve these business capabilities, you will need to be aware of some design guidelines and additional software tools that can cost-effectively achieve the required business functionality.

WEB SITE DESIGN: BASIC BUSINESS CONSIDERATIONS

This is not a text about how to design Web sites. (In Chapter 8, we discuss Web site design issues from a marketing perspective.) Nevertheless, from a business manager's perspective, there are certain design objectives you must communicate to your Web site designers to let them know how you will evaluate their work. At a minimum, your customers will need to find what they need at your site, make a purchase, and leave.

We have all experienced poorly designed e-commerce sites (see Figure 4.17 for a list of the most common consumer complaints about Web sites).

![Figure 4.17: Web Site Features That Annoy Customers](image)

**WEB SITE FEATURES THAT ANNOY CUSTOMERS**

- Links give error reports: 60%
- Too slow to load: 54%
- Forced to download plug-ins: 35%
- Site times out: 33%
- Poorly named links: 31%
- Too many clicks: 28%
- Crowded layout: 24%
- No skip button at introduction: 22%
- No search feature: 20%
- Confusing layout: 20%

**Percentage of Complaints**

Some critics believe poor design is more common than good design. It appears easier to describe what irritates people about Web sites than to describe how to design a good Web site. The worst e-commerce sites make it difficult to find information about their products and make it complicated to purchase goods; they have missing pages and broken links, a confusing navigation structure, and annoying graphics or sounds that you cannot turn off. Table 4.10 restates these negative experiences as positive goals for Web site design.

**Tools for Interactivity and Active Content**

As a manager responsible for building a Web site, you will want to ensure that users can interact with your Web site quickly and easily. As we describe in later chapters, the more interactive a Web site is, the more effective it will be in generating sales and encouraging return visitors.

Although functionality and ease of use are the supreme objectives in site design, you will also want to interact with users and present them with a lively “active” experience. You will want to personalize the experience for customers by addressing their individual needs, and customize the content of your offerings based on their behavior or expressed desires. For example, you may want to offer customers free mortgage calculations or free pension advice, based on their interaction with programs available at your site. In order to achieve these business objectives, you will need to consider carefully the tools necessary to build these capabilities. Simple interactions such as a

<table>
<thead>
<tr>
<th>FACTOR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Functionality</td>
<td>Pages that work, load quickly, and point the customer toward your product offerings</td>
</tr>
<tr>
<td>Informational</td>
<td>Links that customers can easily find to discover more about you and your products</td>
</tr>
<tr>
<td>Ease of use</td>
<td>Simple fool-proof navigation</td>
</tr>
<tr>
<td>Redundant navigation</td>
<td>Alternative navigation to the same content</td>
</tr>
<tr>
<td>Ease of purchase</td>
<td>One or two clicks to purchase</td>
</tr>
<tr>
<td>Multi-browser functionality</td>
<td>Site works with the most popular browsers</td>
</tr>
<tr>
<td>Simple graphics</td>
<td>Avoids distracting, obnoxious graphics and sounds that the user cannot control</td>
</tr>
<tr>
<td>Legible text</td>
<td>Avoids backgrounds that distort text or make it illegible</td>
</tr>
</tbody>
</table>
customer submitting a name, along with more complex interactions involving credit cards, user preferences, and user responses to prompts, all require special programs. Here is a brief description of some commonly used software tools for achieving high levels of site interactivity.

**CGI (Common Gateway Interface)** is a set of standards for communication between a browser and a program running on a server that allows for interaction between the user and the server. CGI permits an executable program to access all the information within incoming requests from clients. The program can then generate all the output required to make up the return page (the HTML, script code, text, etc.), and send it back to the client via the Web server. CGI programs can be written in nearly any programming language as long as they conform to CGI standards. Generally, CGI programs are used with UNIX servers.

For instance, if a user clicks the button “Display the Contents of My Shopping Cart,” the server receives this request and executes a CGI program. The CGI program retrieves the contents of the shopping cart from the database and returns it to the server. The server sends an HTML page that displays the contents of the shopping cart on the user’s screen. Notice all the computing takes place on the server side (this is why CGI programs and others like it are referred to as “server-side” programs).

**Active Server Pages (ASP)** is Microsoft’s version of server-side programming for Windows. Invented by Microsoft in late 1996, ASP has grown rapidly to become the major technique for server-side Web programming in the Windows environment. ASP enables developers to easily create and open records from a database and execute programs within an HTML page, as well as handle all the various forms of interactivity found on e-commerce sites. Like CGI, ASP permits an interaction to take place between the browser and the server. ASP uses the same standards as CGI for communication with the browser. ASP programs are restricted to use on Windows 2003/2000/NT and Windows NT Web servers running Microsoft’s IIS Web server software.

**Java, Java Server Pages (JSP) and JavaScript**

**Java** is a programming language that allows programmers to create interactivity and active content on the client computer—thereby saving considerable load on the server. Java was invented by Sun Microsystems in 1990 as a platform-independent programming language for consumer electronics. The idea was to create a language whose programs (so-called Write Once Run Anywhere (WORA) programs) could operate on any computer regardless of operating system. This would be possible if every operating system (Macintosh, Windows, DOS, UNIX, and mainframe MVS systems) had a Java Virtual Machine (VM) installed that would interpret the Java programs for that environment.

By 1995, it had become clear, however, that Java was more applicable to the Web than to consumer electronics. Java programs (known as **Java applets**) could be downloaded to the client over the Web and executed entirely on the client’s computer. Applet
tags could be included in an HTML page. To enable this, each browser would have to include a Java VM. Today, the leading browsers do include a VM to play Java programs. When the browser accesses a page with an applet, a request is sent to the server to download and execute the program and allocate page space to display the results of the program. Java can be used to display interesting graphics, create interactive environments (such as a mortgage calculator), and directly access the Web server.

Different vendors, including Microsoft, IBM, HP, and others, have produced several versions of the Java language, and even different VMs. Java applets built using Microsoft Java can play well only on Microsoft’s Internet Explorer browser. Therefore, the objective of having Java applets play the same on all Web clients has not succeeded. Many corporations will not allow Java applets through their firewalls for security reasons. Despite the fact that Java applets do not have access to local client system resources (they operate in a “sandbox” for security reasons), IS managers are extremely suspicious of allowing applets served from remote servers to come through the firewall. Many Java applets crash or do not perform well, wasting system resources, and when they do perform, the functions are often trivial (such as flashing logos).

Java Server Pages (JSP), like CGI and ASP, is a Web page coding standard that allows developers to use a combination of HTML, JSP scripts, and Java to dynamically generate Web pages in response to user requests. JSP uses Java “servlets,” small Java programs that are specified in the Web page and run on the Web server to modify the Web page before it is sent to the user who requested it. Java Server Pages are supported by most of the popular application servers on the market today.

JavaScript is a programming language invented by Netscape that is used to control the objects on an HTML page and handle interactions with the browser. It is most commonly used to handle verification and validation of user input, as well as to implement business logic. For instance, JavaScript can be used on customer registration forms to confirm that a valid phone number, zip code, or even e-mail address has been given. Before a user finishes completing a form, the e-mail address given can be tested for validity. JavaScript appears to be much more acceptable to corporations and other environments in large part because it is more stable and also it is restricted to the operation of requested HTML pages.

ActiveX and VBScript

Microsoft—not to be outdone by Sun Microsystems and Netscape—invited the ActiveX programming language to compete with Java and VBScript to compete with JavaScript. When the browser receives an HTML page with an ActiveX control (comparable to a Java applet), the browser simply executes the program. Unlike Java, however, ActiveX has full access to all the client’s resources—printers, networks, hard drives. VBScript performs in the same way as JavaScript. Of course, ActiveX and VBScript work only if you are using Internet Explorer. Otherwise, that part of the screen is blank.
In general, given the conflicting standards for Java, ActiveX, and VBScript and the diversity of user client computers, most e-commerce sites steer clear of these tools. CGI scripts, JSP, and JavaScript are the leading tools for providing active, dynamic content.

**ColdFusion**

ColdFusion is an integrated server-side environment for developing interactive Web applications. Developed by Macromedia, ColdFusion combines an intuitive tag-based scripting language and a tag-based server scripting language (CFML) that lowers the cost of creating interactive features. ColdFusion offers a powerful set of visual design, programming, debugging, and deployment tools.

**PERSONALIZATION TOOLS**

You will definitely want to know how to treat each customer on an individual basis and emulate a traditional face-to-face marketplace. **Personalization** (the ability to treat people based on their personal qualities and prior history with your site) and **customization** (the ability to change the product to better fit the needs of the customer) are two key elements of e-commerce that potentially can make it nearly as powerful as a traditional marketplace, and perhaps even more powerful than direct mail or shopping at an anonymous suburban shopping mall. Speaking directly to the customer on a one-to-one basis, and even adjusting the product to the customer is quite difficult in the usual type of mass marketing, one-size-fits-all commercial transaction that characterizes much of contemporary commerce.

There are a number of methods for achieving personalization and customization. For instance, you could personalize Web content if you knew the personal background of the visitor. You could also analyze the pattern of clicks and sites visited for every customer who enters your site. We discuss these methods in later chapters on marketing. The primary method for achieving personalization and customization is through the placement of cookie files on the user’s client computer. As we discussed in Chapter 3, a cookie is a small text file placed on the user’s client computer that can contain any kind of information about the customer, such as customer ID, campaign ID, or purchases at the site. And then, when the user returns to the site, or indeed goes further into your site, the customer’s prior history can be accessed from a database. Information gathered on prior visits can then be used to personalize the visit and customize the product.

For instance, when a user returns to a site, you can read the cookie to find a customer ID, look the ID up in a database of names, and greet the customer (“Hello Mary! Glad to have you return!”). You could also have stored a record of prior purchases, and then recommend a related product (“How about the wrench tool box now that you have purchased the wrenches?”). And you could think about customizing the
In 1998, Congress amended the federal Rehabilitation Act to require U.S. agencies, government contractors, and others receiving federal money to make electronic and IT services accessible to people with disabilities. Known as Section 508, this legislation requires Web sites of federally funded organizations to be accessible to users who are blind, deaf, blind and deaf, or unable to use a mouse. However, the legislation applies only to U.S. agencies, government contractors, and others receiving federal money, not to the broader e-commerce environment. In 2001, Access Now Inc., an advocacy group for the disabled, sued Southwest Airlines on behalf of more than 50 million disabled Americans for operating a Web site that was inaccessible to the disabled, on the grounds that this violated of the 1990 Americans with Disability Act (ADA).

However, in November 2002, a Federal District Court in Florida, in one of the first court decisions on the applicability of the ADA to Web sites, ruled that ADA applies only to physical spaces, not virtual spaces. However, the judge noted in a footnote that she was surprised that a customer-oriented firm like Southwest Airlines did not “employ all available technologies to expand accessibility to its Web site for visually impaired customers who would be an added source of revenue.”

If designing a Web site to be accessible to the physically challenged is easy to do, as advocates for the disabled argue, why aren’t more Web sites built with accessibility in mind? Part of the problem is that retrofitting an existing site is very difficult, according to online merchants. Building an accessible site from scratch is far easier and far less expensive than the cost of retrofitting an old site.

Today’s graphically rich, multimedia-enabled sites make it more difficult for the approximately 54 million disabled Americans to access Web sites. Blind users, for example, rely on a screen reader that uses software and a speech synthesizer to describe what is on a Web page. But sites that fail to use descriptive tags to explain links or images make it impossible for the screen reader to provide complete information. In the end, only 10% to 20% of all Web sites are accessible to the close to 10 million blind Americans.

Other site design elements make it difficult for the disabled to even navigate through a site. For instance, sites with small or crowded links make it more difficult for people with impaired motor skills to click on a particular link. Sites without closed-captioning cause the hearing impaired to miss built-in audio messages altogether.

There are several simple strategies Web designers can use to improve accessibility. Embedding text descriptions behind images is one example that allows screen readers to announce those descriptions. So instead of saying “Image,” when a screen reader passes over an image, the visually impaired user can hear “Photo of a cruise
ship sitting in a harbor.” Allowing users to set the color and font schemes can also make a difference for the visually impaired. Adding screen magnification tools and sound labels where hyperlinks appear are two additional ways to increase accessibility.

These are examples of “equivalent alternatives” to visual content that disability advocates suggest should be required, both for visual and auditory content, to ensure that individuals with disabilities have equal access to information that appears on-screen. Other guidelines for creating accessible Web sites include ensuring that text and graphics are understandable when viewed without color, using features that enable activation of page elements via a variety of input devices (such as keyboard, head wand, or Braille reader), and providing clear navigation mechanisms (such as navigation bars or a site map) to aid users.

Making page design simpler and easier to follow sounds like good advice for sites trying to appeal to the disabled and well-bodied alike. But should Web sites be required by law to meet everyone’s needs?

**Sources:**
- “Court: Disability Law Doesn’t Apply to the Web,” by Patrick Thibodeau, Computerworld, November 5, 2002.

---

product (“You’ve shown an interest in the elementary training programs for Word. We have a special “How to Study” program for beginners in Office software. Would you like to see a sample copy online?”).

We will describe the use of cookies and their effectiveness in achieving a one-to-one relationship with the customer in Chapter 8, *E-commerce Marketing Communications*.

**The Information Policy Set**

In developing an e-commerce site, you will also need to focus on the set of information policies that will govern the site. You will need to develop a **privacy policy**—a set of public statements declaring to your customers how you treat their personal information that you gather on the site. You will need to establish **accessibility rules**—a set of design objectives that ensure disabled users can effectively access your site. There are more than 40 million Americans who are disabled and require special access routes to buildings as well as computer systems (see *Insight on Society: Design Your Web Site for Accessibility*). E-commerce information policies are described in greater depth in Chapter 9, *Ethical, Social, and Political Issues in E-commerce*. 

**Privacy Policy**

a set of public statements declaring to your customers how you treat their personal information that you gather on the site

**Accessibility Rules**

a set of design objectives that ensure disabled users can effectively access your site
Washington-based Recreational Equipment, Inc. (REI) is the world’s largest online retailer of outdoor gear. Here’s how they succeeded while so many others have failed.

REI is a somewhat unusual company. It was founded in 1938 by Lloyd and Mary Anderson, mountain climbers from Seattle, Washington. The Andersons imported a special ice axe from Austria for themselves and decided to set up a cooperative to help their friends and other fellow outdoor enthusiasts acquire high-quality climbing and camping gear at reasonable prices. Today, REI is the largest consumer cooperative in the United States, with over 2 million members paying a one-time membership fee of $15 that entitles them to an annual dividend equal to about 10% of their annual purchases. An estimated 85% of the company’s earnings are paid out as that member dividend. And the business has grown: Today, REI operates 60 retail stores in 24 states, three online stores, an international mail order operation, and REI Adventures, a travel agency. Kiosks in every store allow customers to access over 78,000 SKUs at either of REI’s Web sites: rei.com and rei-outlet.com. REI employs 6,500 people, generates about $700 million in gross revenues, about $100 million of which comes from the online stores.

REI first started exploring the Internet in the summer of 1995. Netscape had just gone public, and the E-commerce era was just beginning. As with many business success stories, REI’s online venture began with senior executives who recognized the potentially transformative power of the Web, and the mixture of opportunity and possible threat that it represented.

Many traditional “bricks-and-mortar” retailers at that time feared cannibalization of their retail and/or catalog sales if they introduced an online sales outlet. Their nightmare was that starting an online store would merely “steal” their own customers from their regular sales channels. But REI wasn’t deterred. As Dennis Madsen, REI’s president and chief executive officer, said, “We knew that if we could not serve our customers who were looking to shop with us online, they would turn to someone else online. It was never a question for us. Being online meant better serving the customer. Our experience has proven that cannibalization is largely a myth and that our multi-channel customers are our best customers. For instance, dual-channel customers who shopped both online and at stores spent 114% more than single-channel customers. And customers who shop three channels—retail stores, Web, and kiosks—spent 48% more than dual-channel customers.”
REI charged Matt Hyde, who had previously helped start REI Adventures, the company’s travel service, with the mission of launching REI’s first Web site on a budget of approximately $500,000. At the time, Netscape was the only company offering a complete e-commerce suite, so REI chose Netscape’s Merchant Server software installed on an IBM RS/6000 server. And although Hyde recognized that REI was a retailer by trade, not a programming shop, he chose to keep design of the site in-house, using off-the-shelf Web authoring tools rather than outsource creation of the Web site. The rationale: “When [we] took the leap of faith that we could launch this compelling value proposition, and that it could be big, [we] realized we needed to make this a core competency. It couldn’t be outsourced.” The decision was not without its costs, however: Managing rei.com’s growth internally, with no outsourcing, strained REI’s human resources. REI soon discovered that finding people with the requisite skills could be difficult, and even if they could be found, they were a lot more expensive than salespeople. For instance, REI’s Seattle store employs around 400 people, while the company’s online stores have just 90 employees—but the payrolls for the two organizations are about the same.

In September 1996, at a time when few traditional retailers were even looking at online sales, rei.com launched, promoted primarily through direct mail and in-store
notices. The first order arrived 20 minutes later. By February 1997, Hyde and his team knew they were on the right track. Traffic was up by 50% in the two months following Christmas. But that in itself posed a problem. As Hyde remembers, “We chose Netscape early on, and they were clearly the leader [at that time]. But not long after getting the system up, we realized that it was too limited. When you go from a few thousand people checking out your site, to a million every month, you need a lot of infrastructure.”

He also noted, “On the surface, e-commerce sounds relatively easy. It’s not until you have experience trying to integrate a high-volume, high-functionality Web site into existing business processes and applications that you realize that it’s a lot harder than it seems. It’s like an iceberg—the view from the browser is only 10% of what it takes to build a successful and profitable Web site.”

REI had originally hoped to upgrade with Netscape, but, as Hyde said, “that wasn’t working out.” This time they had more of a choice, and looked at offerings from all the major vendors, including Microsoft, IBM, Broadvision, and OpenMarket. “When you change commerce packages, there’s a huge learning curve. I was going to make this change once, but I wasn’t going to do it again, so I wanted to pick the right package . . . for the next several years.” In early 1998, REI decided on IBM’s Net.Commerce server software. An important factor in the decision was IBM’s ability to preserve all the custom coding REI had done over the past two years to connect its online store to its legacy system. “I had hundreds of thousands, if not millions of dollars tied up in this [system], and we didn’t want to throw it away. And since Net.Commerce [would also lessen] the need to do custom coding in the future, it’s a two-fold benefit.”

In August 1998, REI launched a second Web site, rei-outlet.com, using Net.Commerce server software. Once rei-outlet.com was successfully launched, REI then turned to migrating rei.com to the new system, completing the move in October 1998.

In 2002, REI began a third re-building of its Web site by standardizing on a single platform, IBM’s WebSphere, an integrated set of e-commerce site development and operational tools. Prior to this point, REI’s e-commerce infrastructure was a mixture of software applications written both in-house and by a variety of different technology vendors. In the period since its second re-build in 1998, IBM has developed WebSphere as a suite of tools and functionalities built on standards like Java and UNIX that include WebSphere application server, WebSphere Commerce Studio, MQ Series Integrator, IBM VisualAge for Java, and RS6000 pSeries UNIX-based computer servers, which can be configured with multiple processors in order to scale with a business. REI has evidently discovered that it is less costly in the long run to rely on a single vendor like IBM to provide an integrated set of e-commerce applications, rather than building the applications in-house.

Today, rei.com offers more than 78,000 individual items—more than in any of its physical stores—at prices that are the same as in the retail stores; 45,000 pages of in-depth product information; an interactive community system; and a complete adven-
The outlet store, rei-outlet.com, sells merchandise that the company buys specifically for the outlet.

REI’s technology platform continues to rely on 4 IBM RS/6000 servers that link the Web sites and stores to backend legacy databases that are used to store product and customer information. The new WebSphere applications will provide end-to-end business to consumer commerce from shopping and checkout to fulfillment. Orders from the Web site are processed as seamlessly as those from retail stores or mail-order sales.

The system appears to be working well. Forrester’s Multichannel Best Practices Report in 2002 featured REI and Sears as the best multi-channel companies in its U.S. multi-channel retail evaluation, which measures the top 30 things a retailer must do to deliver a great multi-channel experience. REI scored highest for “customer experience” and “technical integration,” with services such as in-store pick-up, no hassle return policies, and 45,000 pages of online product information at REI.com.

With its new focus on an integrated platform from a single vendor, REI plans to implement extensive new customer features utilizing the platform’s capabilities in personalized service and information, as well as more convenience for both online and in-store shoppers. For instance, REI will have the ability to refer new Web customers to nearby stores that are having sales. Or REI could e-mail coupons for bike helmets redeemable at local stores or online to customers who purchased a bicycle. WebSphere could drop an image of hiking boots featured at REI onto the screen of a customer reading an REI “Learn and Share” article on backpacking. Joan Broughton, REI’s vice president of direct sales, says “We’ll be able to gather, integrate and act upon

---

**Sources:**


“REI Named One of Best Multi-channel Retailers,” REI.com press release, December 10, 2002;

“REI Makes Big Bet on WebSphere,” by Richard Karpinski, Internetweek.com, October 18, 2002;

all sorts of meaningful information about our customers in order to improve their experience with REI.*

REI’s clicks-and-bricks strategy has paid off handsomely. In addition to improving relationships with existing customers, the online stores have helped create new customers. About 36% of the online customers are not members of the REI cooperative, meaning they are likely new customers, compared to 15% of REI’s retail customers. And despite management’s early fears, the online stores haven’t cannibalized traditional store sales. In fact, the opposite has occurred: The Web site has attracted new customers and strengthened the relationship with existing customers.

In 2002, REI made sharp gains in operating and net income. Net income doubled to $16 million from $7.5 million. Although comparative store gross sales grew at only 1%, direct sales gained 2.5% and REI.com online sales grew by over 10%. Company officials attribute the sharp rise in operating and net income to a reduction in administrative costs and expenses, as well as a sharp focus on multi-channel retailing, above-expectation sales in newly opened stores, and strong financial management.

Case Study Questions

1. Create a simple logical design and physical design for the REI.com Web site using information provided in the case study, supplemented as necessary by your own research.

2. After reading the case study, identify the key reasons for REI.com’s success thus far.

3. Visit the REI.com Web site and rate its performance on the eight factors listed in Table 4.10 on a scale of 1 to 10 (with 1 being the lowest and 10 the highest). Provide reasons for your ratings.

4. Prepare a short industry analysis of the online outdoor sporting goods and apparel industry. Who are REI’s primary competitors? How well have they developed multi-channel retailing?
4.6 REVIEW

KEY CONCEPTS

- Explain the process that should be followed in building an e-commerce Web site.

Factors you must consider when building an e-commerce site include:
- hardware architecture
- software
- telecommunications capacity
- site design
- human resources
- organizational capabilities

The systems development life cycle (a methodology for understanding the business objectives of a system and designing an appropriate solution) for building an e-commerce Web site involves five major steps:
- Identify the specific business objectives for the site and then develop a list of system functionalities and information requirements.
- Develop a system design specification (both logical design and physical design).
- Build the site, either by in-house personnel or by outsourcing all or part of the responsibility to outside contractors.
- Test the system (unit testing, system testing, and acceptance testing).
- Implement and maintain the site.

The nine basic business and system functionalities an e-commerce site should contain include:
- Digital catalog—allows a site to display goods using text and graphics.
- Product database—provides product information, such as a description, stocking number, and inventory level.
- Customer on-site tracking—enables a site to create a site log for each customer visit, aiding in personalizing the shopping experience and identifying common customer paths and destinations.
- Shopping cart/payment system—provides an ordering system, secure credit-card clearing, and other payment options.
- Customer database—includes customer information such as the name, address, phone number, and e-mail address.
- Sales database—contains information regarding the customer ID, product purchased, date, payment, and shipment to be able to provide after-sale customer support.
- Ad server—tracks the site behavior of prospects and customers that come through e-mail or banner ad campaigns.
- Site tracking and reporting system—monitors the number of unique visitors, pages visited, and products purchased.
- Inventory management system—provides a link to production and suppliers in order to facilitate order replenishment.
Describe the major issues surrounding the decision to outsource site development and/or hosting.

Advantages of building a site in-house include:

- the ability to change and adapt the site quickly as the market demands, and
- the ability to build a site that does exactly what the company needs.

Disadvantages of building a site in-house include:

- the costs may be higher;
- the risks of failure may be greater, given the complexity of issues such as security, privacy, and inventory management;
- the process may be more time-consuming than if you had hired an outside specialist firm to manage the effort; and
- staff may experience a longer learning curve that delays your entry into the market.

Using design templates cuts development time, but preset templates can also limit functionality.

A similar decision is also necessary regarding outsourcing the hosting of the site versus keeping it in-house. Relying on an outside vendor to ensure that the site is live twenty-four hours a day places the burden of reliability on someone else, in return for a monthly hosting fee. The downside is that if the site requires fast upgrades due to heavy traffic, the chosen hosting company may or may not be capable of keeping up. Reliability versus scalability are the issues in this instance.

Identify and understand the major considerations involved in choosing Web server and e-commerce merchant server software.

Early Web sites used single-tier system architecture and consisted of a single-server computer that delivered static Web pages to users making requests through their browsers. The extended functionality of today’s Web sites required the development of a multi-tiered systems architecture, which utilizes a variety of specialized Web servers, as well as links to pre-existing “backend” or “legacy” corporate databases.

All e-commerce sites require basic Web server software to answer requests from customers for HTML and XML pages. When choosing Web server software, companies are also choosing what operating system the site will run on; Apache, which runs on the UNIX system, is the market leader.

Web servers provide a host of services, including:

- processing user HTML requests
- security services
- file transfer protocol
- search engine
- data capture
- e-mail
- site management tools
Dynamic server software allows sites to deliver dynamic content, rather than static, unchanging information. Web application server programs enable a wide range of e-commerce functionality, including creating a customer database, creating an e-mail promotional program, accepting and processing orders, as well as many other services.

E-commerce merchant server software is another important software package that provides catalog displays, information storage and customer tracking, order taking (shopping cart), and credit card purchase processing. E-commerce suites can save time and money, but customization can significantly drive up costs. Factors to consider when choosing an e-commerce suite include its functionality, support for different business models, visual site management tools and reporting systems, performance and scalability, connectivity to existing business systems, compliance with standards, and global and multicultural capability.

- **Understand the issues involved in choosing the most appropriate hardware for an e-commerce site.**

Speed, capacity, and scalability are three of the most important considerations when selecting an operating system, and therefore the hardware that it runs on.

To evaluate how fast the site needs to be, companies need to assess the number of simultaneous users the site expects to see, the nature of their requests, the type of information requested, and the bandwidth available to the site. The answers to these questions will provide guidance regarding the processors necessary to meet customer demand. In some cases, adding additional processing power can add capacity, thereby improving system speed.

Scalability is also an important issue. Increasing processing supply by scaling up to meet demand can be done through:

- **Vertical scaling**—improving the processing power of the hardware, but maintaining the same number of servers;
- **Horizontal scaling**—adding more of the same processing hardware; and
- **Improving processing architecture**—identifying operations with similar workloads and using dedicated tuned servers for each type of load.

- **Identify additional tools that can improve Web site performance.**

In addition to providing a speedy Web site, companies must also strive to have a well-designed site that encourages visitors to buy. Building in interactivity improves site effectiveness, as does personalization techniques that provide the ability to track customers while they are visiting the site. Commonly used software tools for achieving high levels of Web site interactivity and customer personalization include:

- **Common gateway interface (CGI) scripts**—a set of standards for communication between a browser and a program on a server that allows for interaction between the user and the server.
- **Active Server Pages (ASP)**—a Microsoft tool that also permits interaction between the browser and the server.
Java Applets—programs written in the Java programming language that also provide interactivity.

JavaScript—used to validate user input, such as an e-mail address.

ActiveX and VBScript—Microsoft’s version of Java and JavaScript, respectively.

Cookies—text files stored on the user’s hard drive that provide information regarding the user and his or her past experience at a Web site.

**Questions**

1. Name the six main pieces of the e-commerce site puzzle.
2. Define the systems development life cycle and discuss the various steps involved in creating an e-commerce site.
3. Discuss the differences between a simple logical and simple physical Web site design.
4. Why is system testing important? Name the three types of testing and their relation to each other.
5. Compare the costs for system development and system maintenance. Which is more expensive, and why?
6. Why is a Web site so costly to maintain? Discuss the main factors that impact cost.
7. What are the main differences between single-tier and multi-tier site architecture?
8. Name five basic functionalities a Web server should provide.
9. What are the three main factors to consider when choosing the best platform for your Web site?
10. Why is Web server bandwidth an important issue for e-commerce sites?
11. Compare and contrast the various scaling methods. Explain why scalability is a key business issue for Web sites.
12. What are the eight most important factors impacting Web site design, and how do they affect a site’s operation?
13. What are Java and JavaScript? What role do they play in Web site design?
14. Name and describe three tools used to treat customers individually. Why are they significant to e-commerce?
15. What are some of the policies e-commerce businesses must develop before launching a site, and why must they be developed?

**Projects**

1. Go to www.bigstep.com or Yahoo Store (store.yahoo.com). Both sites allow you to create a simple e-tailer Web site for a free trial period. The site should feature at least four pages, including a home page, product page, shopping cart, and contact page. Extra credit will be given for additional complexity and creativity. Come to class prepared to present your e-tailer concept and Web site.
2. Visit several e-commerce sites, not including those mentioned in this chapter, and evaluate the effectiveness of the sites according to the eight basic criteria/functionalities listed in Table 4.10. Choose one site you feel does an excellent job on all the aspects of an effective site and create a presentation, including screen shots, to support your choice.

3. Imagine that you are the head of IT for a fast-growth e-commerce startup. You are in charge of development of the company's Web site. Consider your options for building the site in-house with existing staff, or outsourcing the entire operation. Decide which strategy you believe is in your company's best interest and create a brief presentation outlining your position. Why choose that approach? And what are the estimated associated costs, compared with the alternative? (You'll need to make some educated guesses here—don't worry about being exact.)

4. Choose two of the e-commerce suite software packages listed in Table 4.5 and prepare an evaluation chart that rates the packages on the key factors discussed in the section “Choosing an E-commerce Suite.” Which package would you choose if you were developing a Web site of the type described in this chapter, and why?