[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html" \l "header)

**Chapter**

**7**

**Development Strategies**

* **7.1**[Traditional Versus Web-Based Systems Development](javascript://)
  + **7.1.1**[Traditional Development: In a Traditional Systems Development Environment](javascript://)
  + **7.1.2**[Web-Based Development: in a Web-Based Systems Development Environment](javascript://)
* **7.2**[Evolving Trends](javascript://)
* **7.3**[In-House Software Development Options](javascript://)
  + **7.3.1**[Make or Buy Decision](javascript://)
  + **7.3.2**[Developing Software In-House](javascript://)
  + **7.3.3**[Purchasing a Software Package](javascript://)
  + **7.3.4**[Customizing a Software Package](javascript://)
  + **7.3.5**[Creating User Applications](javascript://)
* **7.4**[Outsourcing](javascript://)
  + **7.4.1**[The Growth of Outsourcing](javascript://)
  + **7.4.2**[Outsourcing Fees](javascript://)
  + **7.4.3**[Outsourcing Issues and Concerns](javascript://)
* **7.5**[Offshoring](javascript://)
* **7.6**[Software as a Service](javascript://)
* **7.7**[Selecting a Development Strategy](javascript://)
  + **7.7.1**[The Systems Analyst’s Role](javascript://)
  + **7.7.2**[Analyzing Cost and Benefits](javascript://)
  + **7.7.3**[Cost-Benefit Analysis Checklist](javascript://)
* **7.8**[The Software Acquisition Process](javascript://)
  + [Step 1: Evaluate the Information System Requirements](javascript://)
  + [Step 2: Identify Potential Vendors or Outsourcing Options](javascript://)
  + [Step 3: Evaluate the Alternatives](javascript://)
  + [Step 4: Perform Cost-Benefit Analysis](javascript://)
  + [Step 5: Prepare a Recommendation](javascript://)
* **7.9**[Completion of Systems Analysis Tasks](javascript://)
  + **7.9.1**[System Requirements Document](javascript://)
  + **7.9.2**[Presentation to Management](javascript://)
  + **7.9.3**[Transition to Systems Design](javascript://)
* **7.10**[Summary](javascript://)
* [Chapter Review](javascript://)
  + [Key Terms](javascript://)

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.1**Traditional Versus Web-Based Systems Development

Just a few years ago, a typical company either developed software itself, purchased a software package (which might need some modification), or hired consultants or outside resources to perform the work. Today, a company has many more choices for software acquisition, including application service providers, web-hosted software options, and firms that offer a variety of enterprise-wide software solutions. This proliferation of choices is due in part due to the enormous changes in business methods and operations made possible by the Internet.

A systems analyst must consider whether development will take place in a traditional environment or in a web-centric framework. There are similarities and differences with both approaches. For example, in an Internet-based system, the web becomes an integral part of the application, rather than just a communication channel, and systems analysts need new application development tools and solutions to handle the new systems.

Two representative web-based development environments are Microsoft’s .NET and the open source MERN stack. Microsoft describes .NET as a developer platform for building and running a variety of application types written in C# and Visual Basic, including web-based, mobile, and traditional desktop applications. The acronym MERN stands for MongoDB, Express, React, and Node. MERN is used to develop universal applications in JavaScript. MongoDB is a database, and Express, React, and Node are libraries or frameworks used for full-stack web development.

Although there is a major trend toward web-based architecture, many firms rely on traditional systems, either because they are using legacy applications that are not easily replaced, or because they do not require a web component to satisfy user needs. To choose between traditional and web-based development, consider some key differences between them. Building the application in a web-based environment can offer greater benefits (and sometimes greater risks) when compared to a traditional environment. The following sections list some characteristics of traditional versus web-based development.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.1.1Traditional Development: In a Traditional Systems Development Environment

* Compatibility issues, including existing hardware and software platforms and legacy system requirements, influence systems design.
* Systems are designed to run on local and wide-area company networks.
* Systems often utilize Internet links and resources, but web-based features are treated as enhancements rather than core elements of the design.
* Development typically follows one of three main paths: in-house development, purchase of a software package with possible modification, or use of outside consultants.
* Scalability can be affected by network limitations and constraints.
* Many applications require substantial desktop computing power and resources.
* Security issues usually are less complex than with web-based systems, because the system operates on a private company network, rather than the Internet.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.1.2Web-Based Development: in a Web-Based Systems Development Environment

* Systems are developed and delivered in an Internet-based framework such as .NET
* Internet-based development treats the web as the platform, rather than just a communication channel.
* Web-based systems are easily scalable and can run on multiple hardware environments.
* Large firms tend to deploy web-based systems as enterprise-wide software solutions for applications such as customer relationship management, order processing, and materials management.
* Web-based software treats the software application as a service that is less dependent on desktop computing power and resources.
* When companies acquire web-based software as a service rather than a product they purchase, they can limit in-house involvement and have the vendor install, configure, and maintain the system by paying agreed-upon fees.
* Web-based software usually requires additional layers, called [**middleware**](javascript://), to communicate with existing software and legacy systems.
* Web-based solutions open more complex security issues that should be addressed.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.2**Evolving Trends

In the constantly changing world of IT, no area is more dynamic than Internet technology. Three examples of evolving trends are Web 2.0, cloud computing, and mobile devices. Systems analysts should be aware of these trends and consider them as they plan large-scale systems.

Many IT professionals use the term [**Web 2.0**](javascript://) to describe a second generation of the web that enables people to collaborate, interact, and share information much more effectively. This new environment is based on continuously available user applications rather than static HTML webpages, without limitations regarding the number of users or how they access, modify, and exchange data. The Web 2.0 environment enhances interactive experiences, including wikis and blogs, and social networking applications such as Facebook, LinkedIn, and Twitter.

As shown in [Figure 7-1](javascript://), the National Institute of Standards and Technology (NIST) defines [**cloud computing**](javascript://) as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.” Cloud computing is often represented by a cloud symbol that indicates a network or the Internet. Cloud computing can be viewed as an online SaaS and data environment supported by powerful computers that makes Web 2.0 possible.

**Figure 7-1**

NIST definition of cloud computing.



Enlarge Image

Source: NIST

[**Mobile devices**](javascript://) have become ubiquitous. Smartphones and tablets are now found in personal use and across the enterprise in most organizations. Today’s mobile devices have enough computing power to provide processing “at the edge,” which means at the end of a network, in the user’s hands. Developing apps for mobile devices requires many new platforms, although many of today’s development tools support web-based and mobile application development at the same time.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.3**In-House Software Development Options

A company can choose to develop its own systems or purchase (and possibly customize) a software package. These development alternatives are shown in [Figure 7-2](javascript://). Although many factors influence this decision, the most important consideration is the total cost of ownership (TCO), which was explained in [Chapter 4](javascript://). In addition to these options, companies also develop user applications designed around commercial software packages, such as Microsoft Office, to improve user productivity and efficiency.

**Figure 7-2**

Instead of outsourcing, a company can choose to develop a system in-house or purchase and possibly customize a commercial package.

Diagram of software components

Description automatically generated

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.3.1Make or Buy Decision

The choice between developing versus purchasing software often is called a [**make or buy**](javascript://), or [**build or buy**](javascript://), decision. The company’s IT department makes, builds, and develops [**in-house software**](javascript://). A [**software package**](javascript://) is obtained from a vendor or an application service provider.

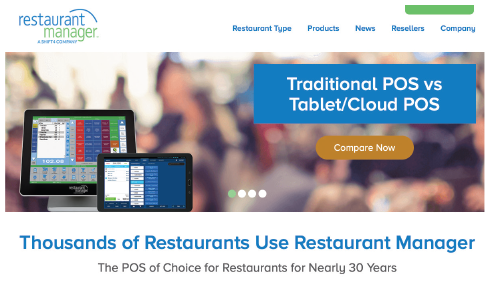
The package might be a standard commercial application or a customized package designed specifically for the purchaser. Companies that develop software for sale are called [**software vendors**](javascript://). A firm that enhances a commercial package by adding custom features and configuring it for a particular industry is called a [**value-added reseller (VAR)**](javascript://).

Software packages are available for every type of business activity. A software package that can be used by many different types of organizations is called a [**horizontal application**](javascript://). An accounting package is a good example of a horizontal application because many different businesses or separate divisions that exist in large, diversified companies can utilize it.

In contrast, a software package developed to handle information requirements for a specific type of business is called a [**vertical application**](javascript://). For example, organizations with special system requirements include colleges, banks, hospitals, insurance companies, construction companies, real estate firms, and airlines. [Figure 7-3](javascript://) shows a typical restaurant point of sale (POS) system running on various devices. The organizations may need vertical applications to handle their unique business requirements but often use horizontal applications for basic business needs, such as payroll processing and accounts payable.

**Figure 7-3**

Restaurants use vertical applications like point of sale (POS) systems to support their unique business requirements.



Source: rmpos

Of the in-house software acquisition options—developing a system, buying a software package, or customizing a software package—each has advantages, disadvantages, and cost considerations, as shown in [Figure 7-4](javascript://). These software acquisition options are described in detail in the following sections.

**Figure 7-4**

Companies consider various factors when comparing in-house development with the purchase of a software package.

| **REASONS FOR IN-HOUSE DEVELOPMENT** | **REASONS FOR PURCHASING A SOFTWARE PACKAGE** |
| --- | --- |
| Satisfy unique business requirements | Lower costs |
| Minimize changes in business procedures and policies | Requires less time to implement |
| Meet constraints of existing systems | Proven reliability and performance benchmarks |
| Meet constraints of existing technology | Requires less technical development staff |
| Develop internal resources and capabilities | Future upgrades provided by the vendor |
| Satisfy unique security requirements | Obtain input from other companies |

Enlarge Table

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.3.2Developing Software In-House

With an enormous variety of software packages available to handle horizontal and vertical business operations, why would a firm choose to develop its own software? Typically, companies choose in-house development to satisfy unique business requirements, minimize changes in business procedures and policies, meet constraints of existing systems and existing technology, and develop internal resources and capabilities.

### Satisfy Unique Business Requirements

Companies often decide to develop software in-house because no commercially available software package can meet their unique business requirements. A college, for example, needs a course scheduling system based on curriculum requirements, student demand, classroom space, and available instructors. A package delivery company needs a system to identify the best combination of routes and loading patterns for the company’s fleet of delivery trucks. If existing software packages cannot handle those requirements, then in-house developed software might be the only choice.

### Minimize Changes in Business Procedures and Policies

A company also might choose to develop its own software if available packages will require changes in current business operations or processes. Installing a new software package almost always requires some degree of change in how a company does business; however, if the installation of a purchased package will be too disruptive, the organization might decide to develop its own software instead.

### Meet Constraints of Existing Systems

Any new software installed must work with existing systems. For example, if a new budgeting system must interface with an existing accounting system, finding a software package that works correctly with the existing accounting system might prove difficult. If so, a company could develop its own software to ensure that the new system will interface with the old system.

### Meet Constraints of Existing Technology

Another reason to develop software in-house is that the new system must work with existing hardware and legacy systems. That could require a custom design, an upgrade to the environment, or in-house software that can operate within those constraints. A systems analyst addresses the issue of technical feasibility during the preliminary investigation. Now, in the systems analysis phase, the analyst must determine whether in-house software development is the best overall solution.

### Develop Internal Resources and Capabilities

By designing a system in-house, companies can develop and train an IT staff who understands the organization’s business functions and information support needs. Many firms feel that in-house IT resources and capabilities provide a competitive advantage because an in-house team can respond quickly when business problems or opportunities arise. For example, if a company lacks internal resources, it must depend on an outside firm for vital business support. Also, outsourcing options might be attractive, but a series of short-term solutions would not necessarily translate into lower TCO over the long term. Top managers often feel more comfortable with an internal IT team to provide overall guidance and long-term stability. In-house development also allows a firm to leverage the skill set of the IT team, which is already on board and being compensated.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.3.3Purchasing a Software Package

If a company decides not to outsource, a commercially available software package might be an attractive alternative to developing its own software. Advantages of purchasing a software package over developing software in-house include lower costs, less time to implement a system, proven reliability and performance benchmarks, less technical development staff, future upgrades that are provided by the vendor, and the ability to obtain input from other companies who already have implemented the software.

### Lower Costs

Because many companies use software packages, software vendors spread the development costs over many customers. Compared with software developed in-house, a software package almost always is less expensive, particularly in terms of initial investment. However, even though the initial cost is less, purchased software can involve expenses caused by business disruption, changing business processes, and retraining employees.

### Requires Less Time to Implement

When a software package is purchased, it already has been designed, programmed, tested, and documented. The in-house time normally spent on those tasks, therefore, is eliminated. Of course, the software must still be installed and integrated into the systems environment, which can take a significant amount of time. Also, even though implementation is quicker, TCO can be higher due to added training expenses and software modifications.

### Proven Reliability and Performance Benchmarks

If the package has been on the market for any length of time, any major problems probably have been detected already and corrected by the vendor. If the product is popular, it almost certainly has been rated and evaluated by independent reviewers.

### Requires Less Technical Development Staff

Companies that use commercial software packages often are able to reduce the number of programmers and systems analysts on the IT staff. Using commercial software also means that the IT staff can concentrate on systems whose requirements cannot be satisfied by software packages.

### Future Upgrades Provided by the Vendor

Software vendors regularly upgrade software packages by adding improvements and enhancements to create a new version or release. A new release of a software package, for example, can include drivers to support a new laser printer or a new type of data storage technology. In many cases, the vendor receives input and suggestions from current users when planning future upgrades.

### Input from Other Companies

Using a commercial software package means that users in other companies can be contacted to obtain their input and impressions. Trying the package or making a site visit to observe the system in operation may be very useful before a final decision is made. Companies can make use of user groups to share experiences with a software package.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.3.4Customizing a Software Package

If the standard version of a software product does not satisfy a company’s requirements, the firm can consider adapting the package to meet its needs. Three ways to customize a software package are as follows:

* Purchase a basic package that vendors will customize to suit the project’s needs. Many vendors offer basic packages in a standard version with add-on components that are configured individually. A vendor offers options when the standard application will not satisfy all customers. A human resources information system is a typical example, because each company handles employee compensation and benefits differently.
* Negotiate directly with the software vendor to make enhancements to meet the project’s needs by paying for the changes.
* Purchase the package and make project-specific modifications, if this is permissible under the terms of the software license. A disadvantage of this approach is that systems analysts and programmers might be unfamiliar with the software and will need time to learn the package and make the modifications correctly.

Additionally, some advantages of purchasing a standard package disappear if the product must be customized. If the vendor does the customizing, the modified package probably will cost more and take longer to obtain. Another issue is future support: Although vendors regularly upgrade their standard software packages, they might not upgrade a customized version. In addition, if the modifications are done by the company purchasing the software, when a new release of the package becomes available, the company might have to modify the new version on its own, because the vendor will not support modifications installed by the customer.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.3.5Creating User Applications

Business requirements sometimes can be fulfilled by a user application, rather than a formal information system or commercial package. User applications are examples of user productivity systems, which were discussed in [Chapter 1](javascript://).

A [**user application**](javascript://) utilizes standard business software, such as Microsoft Word or Microsoft Excel, which has been configured in a specific manner to enhance user productivity. For example, to help a sales rep respond rapidly to customer price requests, an IT support person can set up a form letter with links to a spreadsheet that calculates incentives and discounts. In addition to configuring the software, the IT staff can create a [**user interface**](javascript://), which includes screens, commands, controls, and features that enable users to interact more effectively with the application.

In some situations, user applications offer a simple, low-cost solution. Most IT departments have a backlog of projects, and IT solutions for individuals or small groups do not always receive a high priority. At the same time, application software is more powerful, flexible, and user-friendly than ever. Companies such as Apple and Microsoft offer software suites and integrated applications that can exchange data with programs that include tutorials, wizards, and Help features to guide less-experienced users who know what they need to do but do not know how to make it happen.

Many companies empower lower-level employees by providing more access to data and more powerful data management tools. The main objective is to allow lower-level employees more access to the data they require to perform their jobs, with no intervention from the IT department. This can be accomplished by creating effective user interfaces for company-wide applications, such as accounting, inventory, and sales systems. Another technique is to customize standard productivity software, such as Microsoft Word or Microsoft Excel, to create user applications. In either case, empowerment makes the IT department more productive because it can spend less time responding to the daily concerns and data needs of users and more time on high-impact systems development projects that support strategic business goals.

Empowerment reduces costs and makes good business sense, but companies that adopt this approach must provide the technical support that empowered users require. In most large- and medium-sized companies, a [**service desk**](javascript://) within the IT department is responsible for providing user support. The service desk offers services such as hotline assistance, training, and guidance to users who need technical help.

Users typically require spreadsheets, database management programs, and other software packages to meet their information needs. If user applications access corporate data, appropriate controls must be provided to ensure data security and integrity. For example, some files should be hidden totally from view; others should have read-only properties, so users can view, but not change, the data.

**Case in Point 7.1**

### Doug’s Sporting Goods

* Doug’s Sporting Goods sells hiking and camping supplies. The company has grown considerably in the past two years. They want to develop a customer order entry system and hired your IT consulting firm to advise them about development strategies. They are leaning toward in-house development because they do not want to depend on outside vendors and suppliers for technical support and upgrades. They also say they are not interested in selling on the Web, but that could change in the future. They want to meet with you tomorrow to make a decision. What will you say to them at the meeting?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.4**Outsourcing

[**Outsourcing**](javascript://) is the transfer of information systems development, operation, or maintenance to an outside firm that provides these services, for a fee, on a temporary or long-term basis. Outsourcing can refer to relatively minor programming tasks: renting software from a service provider, outsourcing a basic business process (often called [**business process outsourcing**](javascript://), or **BPO**), or handling a company’s entire IT function.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.4.1The Growth of Outsourcing

Traditionally, firms outsourced IT tasks as a way of controlling costs and dealing with rapid technological change. Oracle cites data that shows that businesses spend up to 80% of their IT budgets maintaining existing software and systems, which forces IT managers “. . . to spend time managing tedious upgrades instead of revenue-generating IT projects.” While those reasons still are valid, outsourcing has become part of an overall IT strategy for many organizations. The outsourcing trend also has affected software vendors, who have adjusted their marketing accordingly.

A firm that offers outsourcing solutions is called a [**service provider**](javascript://). Some service providers concentrate on specific software applications; others offer business services such as order processing and customer billing. Still others offer enterprise-wide software solutions that integrate and manage functions such as accounting, manufacturing, and inventory control.

Two popular outsourcing options involve application service providers and firms that offer Internet business services. These terms are explained in the following sections.

### Application Service Providers

An [**application service provider (ASP)**](javascript://) is a firm that delivers a software application, or access to an application, by charging a usage or subscription fee. An ASP provides more than a license to use the software; it rents an operational package to the customer. ASPs typically provide commercially available software such as databases and accounting packages. If a company uses an ASP to supply a data management package, for example, the company does not have to design, develop, implement, or maintain the package.

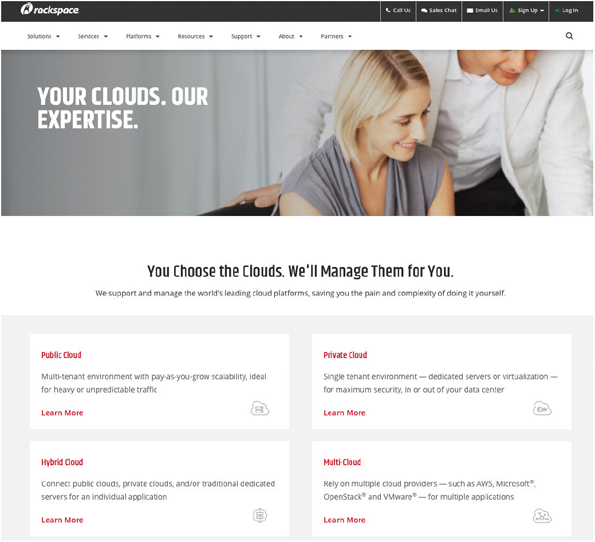
### Internet Business Services

Some firms offer [**Internet business services (IBSs)**](javascript://), which provide powerful web-based support for transactions such as order processing, billing, and customer relationship management. Another term for IBS is [**managed hosting**](javascript://), because the outside firm (host) manages system operations.

An IBS solution is attractive to customers because it offers online data center support, mainframe computing power for mission-critical functions, and universal access via the Internet. Many firms, such as Rackspace, compete in the managed cloud services market, as shown in [Figure 7-5](javascript://).

**Figure 7-5**

Rackspace managed cloud services.



Enlarge Image

Source: Rackspace

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.4.2Outsourcing Fees

Firms that offer Software as a Service, rather than a product, have developed fee structures that are based on how the application is used by customers during a specific time period. Several models exist, including fixed fee, subscription, and usage or transaction. A [**fixed fee model**](javascript://) uses a set fee based on a specified level of service and user support. A [**subscription model**](javascript://) has a variable fee based on the number of users or workstations that have access to the application. Finally, a [**usage model**](javascript://) or [**transaction model**](javascript://) charges a variable fee based on the volume of transactions or operations performed by the application.

When a company considers outsourcing, it should estimate usage characteristics to determine which fee structure would be most desirable and then attempt to negotiate a service provider contract based on that model.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.4.3Outsourcing Issues and Concerns

When a company decides to outsource IT functions, it takes an important step that can affect the firm’s resources, operations, and profitability. Mission-critical IT systems should be outsourced only if the result is a cost-attractive, reliable business solution that fits the company’s long-term business strategy and involves an acceptable level of risk. Moving IT work overseas raises even more issues, including potential concerns about control, culture, communication, and security.

In addition to long-term strategic consequences, outsourcing also can raise some concerns. For example, a company must turn over sensitive data to an external service provider and trust the provider to maintain security, confidentiality, and quality. Also, before outsourcing, a company must carefully review issues relating to insurance, potential liability, licensing and information ownership, warranties, and disaster recovery.

Most important, a company considering outsourcing must realize that the solution can be only as good as the outsourcing firm that provides the service. A dynamic economy can give rise to business failures and uncertainty about the future. In this climate, it is especially important to review the history and financial condition of an outsourcing firm before making a commitment.

Mergers and acquisitions also can affect outsourcing clients. Even with large, financially healthy firms, a merger or acquisition can have some impact on clients and customers. If stability is important, an outsourcing client should consider these issues.

Outsourcing can be especially attractive to a company whose volume fluctuates widely, such as a defense contractor. In other situations, a company might decide to outsource application development tasks to an IT consulting firm if the company lacks the time or expertise to handle the work on its own. Outsourcing relieves a company of the responsibility of adding IT staff in busy times and downsizing when the workload lightens. A major disadvantage of outsourcing is that it raises employee concerns about job security. Talented IT people usually prefer positions where the firm is committed to in-house IT development—if they do not feel secure, they might decide to work directly for the service provider.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

# 7.5Offshoring

[**Offshoring**](javascript://), also known as [**offshore outsourcing**](javascript://) or [**global outsourcing**](javascript://), refers to the practice of shifting IT development, support, and operations to other countries. In a trend similar to the outflow of manufacturing jobs over a several-decade period, many firms are sending IT work overseas.

IT work can move offshore even faster than manufacturing, because it is easier to ship work across networks and put consultants on airplanes than it is to ship bulky raw materials, build factories, and deal with tariffs and transportation issues. Several years ago, the IT consulting firm Gartner, Inc., accurately forecasted the steady growth of offshore outsourcing and predicted that outsourcing would evolve from labor-intensive maintenance and support to higher-level systems development and software design.

The main reason for offshoring is the same as domestic outsourcing: lower bottom-line costs. Offshore outsourcing, however, involves some unique risks and concerns. For example, workers, customers, and shareholders in some companies have protested this trend and have raised public awareness of possible economic impact. Even more important, offshore outsourcing involves unique concerns regarding project control, security issues, disparate cultures, and effective communication with critical functions that might be located halfway around the globe.

**Case in Point 7.2**

### Turnkey Services

* Turnkey Services is an ASP that offers payroll and tax preparation services for hundreds of businesses in the Midwest. The firm is considering a major expansion into accounting and financial services and is looking into the possibility of supporting this move by hiring IT subcontractors in several foreign countries. Turnkey’s president has asked you to help him reach a decision. Specifically, he wants you to cite the pros and cons of offshoring. He wants you to present your views at a meeting of Turnkey managers next week. How will you proceed?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.6**Software as a Service

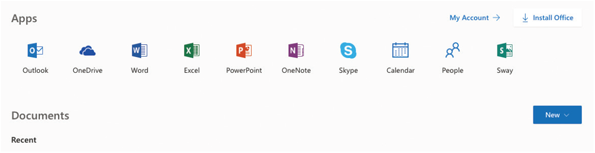
In the traditional model, software vendors develop and sell application packages to customers. Typically, customers purchase licenses that give them the right to use the software under the terms of the license agreement. Although this model still accounts for a large percentage of software acquisition, a new model, called [**Software as a Service (SaaS)**](javascript://), is changing the picture dramatically.

SaaS is a model of software deployment in which an application is hosted as a service provided to customers over the Internet. SaaS reduces the customer’s need for software maintenance, operation, and support. In effect, SaaS provides the functionality the customer needs, but without the associated development, infrastructure, and maintenance costs.

In a highly competitive marketplace, major vendors constantly strive to deliver new and better solutions. For example, Microsoft claims that its SaaS platform offers the best solution and business value. One of their more popular consumer SaaS offerings is Office 365, shown in [Figure 7-6](javascript://). This is a full-fledged version of the Microsoft Office suite that runs in a browser window.

**Figure 7-6**

Microsoft Office 365 provides web-based access to the complete Office suite.



Enlarge Image

**Source:** Microsoft Corporation

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.7**Selecting a Development Strategy

Selecting the best development strategy is an important decision that requires companies to consider multiple factors. The systems analyst has an important role to play in this decision-making process. In particular, analyzing the costs and benefits of each development alternative is a key to providing objective data to management. A cost-benefit checklist can help guide this analysis.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.7.1The Systems Analyst’s Role

At some point in the systems development process, the company must decide whether to use an outsourcing option, develop software in-house, acquire a software package, develop user applications, or select some combination of these solutions. The decision depends on the company’s current and anticipated future needs. It will affect the remaining SDLC phases and the systems analyst’s subsequent involvement in the project. The decision to develop software in-house, for example, will require more participation from the systems analyst than outsourcing or choosing a commercial package. Management usually makes a determination after receiving written recommendations from the IT staff and a formal presentation, which is described later in this chapter.

Even a single system can use a mix of software alternatives. For example, a company might purchase a standard software package to process its payroll and then develop its own software to handle the interface between the payroll package and the company’s in-house manufacturing cost analysis system.

The evaluation and selection of alternatives is not a simple process. The objective is to obtain the product with the lowest TCO, but actual cost and performance can be difficult to forecast. When selecting hardware and software, systems analysts often work as an [**evaluation and selection team**](javascript://). A team approach ensures that critical factors are not overlooked and that a sound choice is made. The evaluation and selection team also must include users, who will participate in the selection process and feel a sense of ownership in the new system.

The primary objective of the evaluation and selection team is to eliminate system alternatives that will not meet requirements, rank the alternatives that are feasible, and present the viable alternatives to management for a final decision. The process begins with a careful study of the costs and benefits of each alternative, as explained in the following section.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.7.2Analyzing Cost and Benefits

Financial analysis tools have been around for a long time. From the abacus to the pocket calculator, people have always sought easier ways to work with numbers. This section describes cost and benefit analysis and explains popular tools that can help a systems analyst examine an IT project.

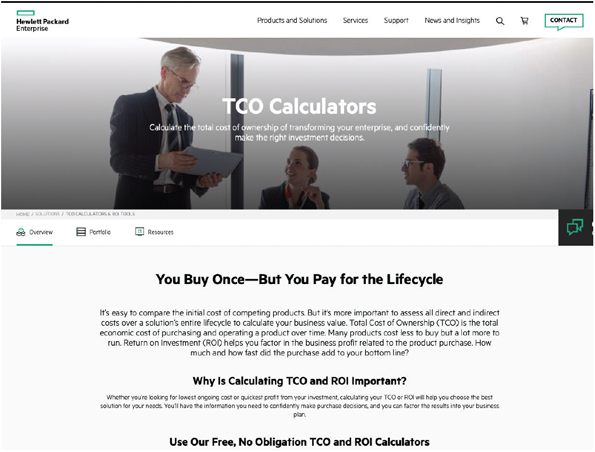
[Chapter 2](javascript://) explained that economic feasibility is one of the four feasibility measurements that are made during the preliminary investigation of a systems request. Now, at the end of the systems analysis phase of the SDLC, financial analysis tools and techniques must be applied to evaluate development strategies and decide how the project will move forward. [Part C](javascript://) of the Systems Analyst’s Toolkit describes three popular tools, which are payback analysis, return on investment (ROI), and net present value (NPV). [**Payback analysis**](javascript://) determines how long it takes an information system to pay for itself through reduced costs and increased benefits. [**Return on investment (ROI)**](javascript://) is a percentage rate that compares the total net benefits (the return) received from a project to the total costs (the investment) of the project. The [**net present value (NPV)**](javascript://) of a project is the total value of the benefits minus the total value of the costs, with both costs and benefits adjusted to reflect the point in time at which they occur.

These tools, and others, can be used to determine TCO, which was described in [Chapter 4](javascript://). At this stage, the analyst will identify specific systems development strategies and choose a course of action. For example, a company might find that its TCO will be higher if it develops a system in-house, compared with outsourcing the project or using an ASP.

An accurate forecast of TCO is critical, because nearly 80% of total costs occur after the purchase of the hardware and software, according to Gartner, Inc. An IT department can develop its own TCO estimates or use TCO calculation tools offered by vendors. For example, as shown in [Figure 7-7](javascript://), HP Enterprise offers several free TCO calculators to determine the ROI of various development strategies and migration options.

**Figure 7-7**

HP Enterprise provides several free TCO calculators.



Enlarge Image

**Source:** Hewlett Packard(HP)

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.7.3Cost-Benefit Analysis Checklist

[Chapter 2](javascript://) explained how to use the payback analysis tool during the preliminary investigation to help determine whether a project is economically feasible. Now, all the financial analysis tools will be used to evaluate various development strategies. The best way to apply the tools is to develop a cost-benefit checklist with the following steps:

* List each development strategy being considered.
* Identify all costs and benefits for each alternative. Be sure to indicate when costs will be incurred and benefits realized.
* Consider future growth and the need for scalability.
* Include support costs for hardware and software.
* Analyze various software licensing options, including fixed fees and formulas based on the number of users or transactions.
* Apply the financial analysis tools to each alternative.
* Study the results and prepare a report to management.

**Case in Point 7.3**

### Sterling Associates

* Sterling Associates specializes in advising clients on IT projects and information systems development. Marketing is creating a brochure for prospective new clients, and they want you to develop a section that describes payback analysis, ROI, and NPV in simple terms and mentions the pros and cons of each financial analysis tool. How do you proceed?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.8**The Software Acquisition Process

Although each situation is different, the following section describes a typical example of the issues and tasks involved in software acquisition.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## Step 1: Evaluate the Information System Requirements

Based on the analysis of the system requirements, the system’s key features must be identified; network and web-related issues considered; volume and future growth estimated; any hardware, software, or personnel constraints specified; and an RFP or a quotation prepared.

### Identify Key Features

Whether in-house development or outsourcing options are being considered, the analyst must develop a clear, detailed list of features that can serve as an overall specification for the system. Using the data gathered during fact-finding, which was discussed in [Chapter 4](javascript://), list all system requirements and critical features. This information will be included in the **system requirements document**, which is the end product of the SDLC systems analysis phase.

### Consider Network and Web-Related Issues

As the system requirements are evaluated, the network and web-related issues must be considered. The analyst must decide whether the system will run on a network, the Internet, or a company intranet and build these requirements into the design. Also, it must be determined whether the system will exchange data with vendor or customer systems and ensure that the system will be compatible.

### Estimate Volume and Future Growth

The analyst needs to know the current volume of transactions and forecast future growth. [Figure 7-8](javascript://) shows volume estimates for an order processing system. In addition to current levels, the figure displays two forecasts; one based on the existing order processing procedures and another that assumes a new website is operational.

**Figure 7-8**

Volume estimates for an order processing system showing current activity levels and two forecasts: one based on the existing order processing procedures and another that assumes a new website is operational.

### Online Order Processing System Estimated Activity during Next 12-Month Period

|  |  |  |  |
| --- | --- | --- | --- |
|  | **CURRENT LEVEL** | **FUTURE GROWTH** (based on existing procedures) | **FUTURE GROWTH** (assuming new website is operational) |
| Customers | 36,500 | 40,150 | 63,875 |
| Daily Orders | 1,435 | 1,579 | 2,811 |
| Daily Order Lines | 7,715 | 7,893 | 12,556 |
| Sales Reps | 29 | 32 | 12 |
| Order Processing Support Staff | 2 | 4 | 3 |
| Products | 600 | 650 | 900 |

Enlarge Table

A comparison of the two forecasts shows that the website will generate more new customers, process almost 80% more orders, and substantially reduces the need for sales reps and support staff. If in-house development is being considered, make sure that the software and hardware can handle future transaction volumes and data storage requirements. Conversely, if outsourcing is being considered, volume and usage data is essential to analyze ASP fee structures and develop cost estimates for outsourcing options.

### Specify Hardware, Software, or Personnel Constraints

The analyst must determine whether existing hardware, software, or personnel issues will affect the acquisition decision. For example, if the firm has a large number of legacy systems or if an Enterprise Resource Planning (ERP) strategy has been adopted, these factors will have an impact on the decision. Also, the company’s policy regarding outsourcing IT functions must be investigated, and whether outsourcing is part of a long-term strategy. With regard to personnel issues, in-house staffing requirements must be defined to develop, acquire, implement, and maintain the system—and determine whether the company is willing to commit to those staffing levels versus an outsourcing option.

### Prepare a Request for Proposal or Quotation

To obtain the information needed to make a decision, the analyst should prepare an RFP or an RFQ. The two documents are similar but used in different situations, based on whether or not a specific software product has been selected.

A [**request for proposal (RFP)**](javascript://) is a document that describes the company, lists the IT services or products needed, and specifies the features required. An RFP helps ensure that the organization’s business needs will be met. An RFP also spells out the service and support levels required. Based on the RFP, vendors can decide if they have a product that will meet the company’s needs. RFPs vary in size and complexity, just like the systems they describe. An RFP for a large system can contain dozens of pages with unique requirements and features. An RFP can be used to designate some features as essential and others as desirable. An RFP also requests specific pricing and payment terms. There are several online sources where RFP templates can be found.

When several responses to an RFP are evaluated, it is helpful to use an evaluation model. An [**evaluation model**](javascript://) is a technique that uses a common yardstick to measure and compare vendor ratings.

[Figure 7-9](javascript://) shows two evaluation models for a network project. The evaluation model at the top of the figure simply lists the key elements and each vendor’s score. The model at the bottom of the figure adds a weight factor. In this example, each element receives a rating based on its relative importance. Although the initial scores are the same in both models, note that vendor A has the highest point total in the top example, but vendor C emerges as the best in the weighted model.

**Figure 7-9**

The three vendors have the same initial ratings, but the two evaluation models produce different results. In the unweighted model at the top of the figure, vendor A has the highest total points. However, after applying weighting factors, vendor C is the winner, as shown in the model at the bottom of the figure.

### Unweighted Evaluation Model for a Network Project

|  |  |  |  |
| --- | --- | --- | --- |
| **Instructions:** Rate each vendor on a scale from 1(low) to 10 (high), then add vendor scores to calculate total points. | | | |
|  | **VENDOR A** | **VENDOR B** | **VENDOR C** |
| **Price** | 6 | 5 | 9 |
| **Completion Date** | 2 | 5 | 8 |
| **Layout/Design** | 8 | 8 | 5 |
| **References** | 10 | 6 | 3 |
| **TOTAL POINTS** | 26 | 24 | 25 |

Enlarge Table

### Weighted Evaluation Model for a Network Project

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instructions:** Rate each vendor on a scale from 1(low) to 10 (high), then multiply the vendor’s score by the weight factor. Add vendor scores to calculate total points. | | | | |
|  | **WEIGHT FACTOR** | **VENDOR A** | **VENDOR B** | **VENDOR C** |
| **Price** | 25 |  |  |  |
| **Completion Date** | 25 |  |  |  |
| **Layout/Design** | 35 |  |  |  |
| **References** | 15 |  |  |  |
| **TOTAL POINTS** | 100 | 630 | 620 | 645 |

Enlarge Table

No standard method exists for assigning the weight factors. Each firm will have its own approach, which might be tailored to fit a specific situation. An analyst usually obtains as much input as possible and then circulates proposed values for further comment and, hopefully, a consensus.

Evaluation models are valuable tools that can be used throughout the SDLC. A spreadsheet program can be used to build an evaluation model, experiment with different weighting factors, and graph the results.

A [**request for quotation (RFQ)**](javascript://) is more specific than an RFP. When an RFQ is used, the specific product or service desired is already known; only price quotations or bids are needed. RFQs can involve outright purchase or a variety of leasing options and can include maintenance or technical support terms. Some vendors even provide convenient RFP or RFQ forms on their websites. RFPs and RFQs have the same objective: to obtain vendor replies that are clear, comparable, and responsive, so that a well-informed selection decision can be made.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## Step 2: Identify Potential Vendors or Outsourcing Options

The next step is to identify potential vendors or outsourcing providers. The Internet is a primary marketplace for all IT products and services, and descriptive information can be found on the web about all major products and acquisition alternatives.

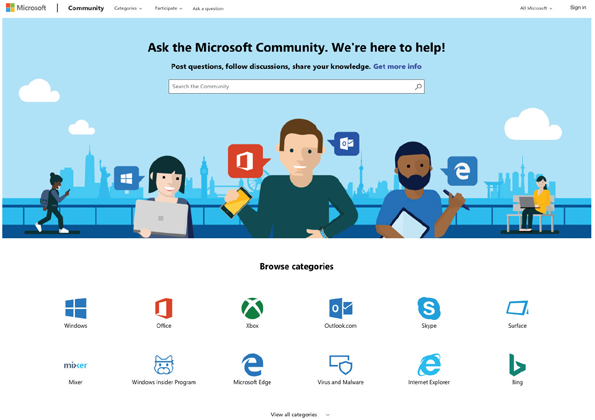
If vertical applications for specific industries need to be located, industry trade journals or websites can be used to find reviews for industry-specific software. Industry trade groups can often provide referrals to companies that offer specific software solutions.

Another approach is to work with a consulting firm. Many IT consultants offer specialized services that help companies select software packages. A major advantage of using a consultant is that the analyst can tap into broad experience that is difficult for any one company to acquire. Consultants can be located by contacting professional organizations or industry sources or simply by searching the Internet. Using a consultant involves additional expense but can prevent even more costly mistakes.

No matter what the topics of interest are, there are sure to be one or more online [**forums**](javascript://) where people gather to meet, offer support, and exchange ideas. Forums can be hosted by private or public entities, or reside in larger communities such as Google Groups or Reddit, which allow users to join existing groups or start their own. A web search can locate forums of interest, or the websites of specific companies, such as Microsoft, and can provide a valuable source of information for IT professionals, including blogs, forums, webcasts, and other resources, as shown in [Figure 7-10](javascript://).

**Figure 7-10**

Microsoft Community is a valuable resource for IT professionals.



Enlarge Image

**Source:** Microsoft Corporation

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## Step 3: Evaluate the Alternatives

After identifying the alternatives, the analyst must select the one that best fits the company’s needs. Information about the options should be obtained from as many sources as possible, including vendor presentations and literature, product documentation, trade publications, and companies that perform software testing and evaluation. To learn more about particular software packages, search the Internet using keywords that describe the application. Websites maintained by consultants and software publishers often include product references and links to vendors. As part of the evaluation process, try to obtain information from existing users, test the application, and benchmark the package.

### Existing Users

Existing users can be contacted to obtain feedback and learn about their experiences. For large-scale software packages, ASPs and vendors typically supply user references. User references are important because it must be known whether the software package has worked well for similar companies. Be aware that some vendors limit their reference lists to satisfied clients, so mostly positive feedback should be expected from those firms.

### Application Testing

If a software package is one of the options, find out if it is possible for users in the organization to try the product. For horizontal applications or small systems, using a demo copy to enter a few sample transactions could be an acceptable test. For vertical applications or large systems, a team of IT staff and users might need several days or weeks to perform tests.

### Benchmarking

To determine whether a package can handle a certain transaction volume efficiently, a benchmark test can be performed. A [**benchmark**](javascript://) measures the time a package takes to process a certain number of transactions. For example, a benchmark test can measure the time needed to post 1,000 sales transactions.

If benchmarks are used, remember that a benchmark test is conducted in a controlled environment, which might not resemble the actual day-to-day situation at the project’s company. Although benchmarking cannot predict project-specific results, benchmark testing is a good way to measure relative performance of two or more competing products in a standard environment.

Many IT publications publish regular reviews of individual packages, including benchmark tests, and often have annual surveys covering various categories of software. Most of the publications now offer online and mobile versions, with additional features, search capability, and IT links.

Information can also be obtained from independent firms that benchmark various software packages and sell comparative analyses of the results, as shown in [Figure 7-11](javascript://). The Transaction Processing Performance Council (TPC) is an example of a nonprofit organization that publishes standards and reports for its members and the general public.

**Figure 7-11**

The Transaction Processing Performance Council (TPC) is a nonprofit organization that publishes standards and reports for its members and the general public.



Enlarge Image

**Source:** Transaction Processing Performance Council (TPC)

Finally, each package should be matched against the RFP features and rank the choices. If some features are more important than others, give them a higher weight using an evaluation model similar to the one shown in [Figure 7-11](javascript://).

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## Step 4: Perform Cost-Benefit Analysis

Review the suggestions in this chapter and in [Part C](javascript://) of the Systems Analyst’s Toolkit to develop a spreadsheet to identify and calculate TCO for each option being considered. Be sure to include all costs, using the volume forecasts prepared. If outsourcing options are being considered, carefully study the alternative fee structure models described earlier. If possible, prepare charts to show the results graphically, and build in what-if capability so the impact of one or more variables changing can be gauged.

If a software package is being considered, be sure to consider acquisition options. When software is purchased, a [**software license**](javascript://) is being bought that gives the purchaser the right to use the software under certain terms and conditions. For example, the license could allow the software to be used only on a single computer, a specified number of computers, a network, or an entire site, depending on the terms of the agreement. Other license restrictions could prohibit making the software available to others or modifying the program. For desktop applications, software license terms and conditions usually cannot be modified. For large-scale systems, license agreement terms often can be negotiated.

Also consider user support issues, which can account for a significant part of TCO. If an outsourcing alternative is selected, the arrangement probably will include certain technical support and maintenance. If in-house development is chosen, the cost of providing these services must be considered. If a software package is purchased, consider a supplemental [**maintenance agreement**](javascript://), which offers additional support and assistance from the vendor. The agreement might provide full support for a period of time or list specific charges for particular services. Some software packages provide free technical support for a period of time. Afterward, support is offered with a charge per occurrence, or per minute or hour of technical support time. Some software vendors contact registered owners whenever a new release is available and usually offer the new release at a reduced price.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## Step 5: Prepare a Recommendation

The analyst should prepare a recommendation that evaluates and describes the alternatives, together with the costs, benefits, advantages, and disadvantages of each option. At this point, it may be required to submit a formal system requirements document and deliver a presentation. Review the suggestions for presenting written proposals and oral presentations in [Part A](javascript://) of the Systems Analyst’s Toolkit. Additional suggestions about preparing the system requirements document and the management presentation are contained in the following section.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.9**Completion of Systems Analysis Tasks

To complete the systems analysis phase, the analyst must finalize the system requirements document, present their findings to management, and begin the transition to systems design.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.9.1System Requirements Document

The **system requirements document** contains the requirements for the new system, describes the alternatives that were considered, and makes a specific recommendation to management. This important document is the starting point for measuring the performance, accuracy, and completeness of the finished system before entering the systems design phase.

The system requirements document is like a contract that identifies what the system developers must deliver to users. Recall that system requirements are identified during the fact-finding process, and a system requirements checklist is created at that time. Various examples of system requirements are listed in [Chapter 4](javascript://). The system requirements document should be written in language that users can understand so they can offer input, suggest improvements, and approve the final version.

Because the system requirements document can be lengthy, they should be formatted and organized, so it is easy to read and use. The system requirements document should include a cover page and a detailed table of contents. An index and a glossary of terms can be added to make the document easier to use. The content of the system requirements document will depend on the company and the complexity of the system.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.9.2Presentation to Management

The presentation to management at the end of the systems analysis phase is one of the most critical milestones in the systems development process. At this point, managers make key decisions that affect the future development of the system.

Prior to a management presentation, two other presentations may be given: one to the principal individuals in the IT department to keep them posted, and another presentation to users to answer their questions and invite feedback. The system requirements document is the basis for all three presentations, and it (or a summary) should be distributed in advance so the recipients can review it.

When preparing the presentation, review the suggestions in [Part A](javascript://) of the Systems Analyst’s Toolkit, which can help design and deliver a successful presentation. If a slide presentation is planned, review the Toolkit guidelines for effective presentations. In addition to the techniques found in the Toolkit, also keep the following suggestions in mind:

* Begin the presentation with a brief overview of the purpose and primary objectives of the system project, the objectives of this presentation, and what decisions need to be made.
* Summarize the primary viable alternatives. For each alternative, describe the costs, advantages, and disadvantages.
* Explain why the evaluation and selection team chose the recommended alternative.
* Allow time for discussion and for questions and answers.
* Obtain a final decision from management or agree on a timetable for the next step in the process.

The object of the management presentation is to obtain approval for the development of the system and to gain management’s full support, including necessary financial resources. Management probably will choose one of five alternatives: develop an in-house system, modify a current system, purchase or customize a software package, perform additional systems analysis work, or stop all further work. Depending on their decision, the next task of the systems analyst will be one of the following:

* Implement an outsourcing alternative. If outsourcing is selected, the analyst will work with representatives of the service provider to achieve a smooth transition to the new environment.
* Develop an in-house system. Begin systems design tasks, as described in [Chapters 8](javascript://), [9](javascript://), and [10](javascript://).
* Purchase or customize a software package. Negotiate the purchase terms with the software vendor for management approval. Then, if the package will be used without modification, the analyst can begin planning the systems implementation phase. If modifications must be made to the package, the next step is to start the systems design phase. If the vendor will make the modifications, then the analyst’s next step is to start planning the testing and documentation of the modifications as part of the systems implementation phase, which is described in [Chapter 11](javascript://).
* Perform additional systems analysis work. Management might want the analyst to investigate certain alternatives further, explore alternatives not examined, develop a prototype, reduce the project scope because of cost constraints, or expand the project scope based on new developments. If necessary, the analyst will perform the additional work and schedule a follow-up presentation.
* Stop all further work. The decision might be based on the analyst’s recommendation, a shift in priorities or costs, or for other reasons. Whatever the reason, if that is management’s decision, then there are no additional tasks for the project other than to file all the research in a logical location, so it can be retrieved if the project is reopened in the future.

After the presentation and management decision, the project will begin a transition to the systems design phase of the SDLC. If an in-house system is being developed, or a package is being modified, a model of the proposed system will be built, and the analyst will start designing the user interface, output, input, and data structures.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

## 7.9.3Transition to Systems Design

In a traditional SDLC environment, systems design usually started when the systems analysis phase was done. Using the system requirements specification as a blueprint, developers transformed the logical design into a working model that could be tested, reviewed by users, and implemented. Today, the process is much more dynamic. In general, systems development is faster, more flexible, and more user oriented. The introduction of agile development has changed the landscape significantly. Depending on the project, system developers often blend traditional and cutting-edge development methods, because what works in one situation might not work in another.

This textbook discusses systems analysis in [Chapters 4](javascript://), [5](javascript://), [6](javascript://), and [7](javascript://) and systems design in [Chapters 8](javascript://), [9](javascript://), and [10](javascript://). However, in a typical IT workplace, all these tasks—and more—are integrated and managed together.

Regardless of the development method, systems design requires accurate documentation. Traditionally, a system requirements document provided detailed specifications for output, input, data, processes, and whatever else was needed. Although agile methods do not require a particular form of documentation, a successful development team must understand and record user requirements as they evolve during the project.

A [**logical design**](javascript://) defines what must take place, not how it will be accomplished. Logical designs do not address the actual methods of implementation. In contrast, a [**physical design**](javascript://) is like a set of blueprints for the actual construction of a building. Typically, a physical design describes the actual processes of entering, verifying, and storing data; the physical layout of data files and sorting procedures; the format of reports; and so on. Because logical and physical designs are related so closely, good systems design is impossible without careful, accurate systems analysis. For example, the analyst might return to fact-finding if it was discovered that an important issue was overlooked, if users had significant new needs, or if legal or governmental requirements changed.

### A Question of Ethics

* [iStock.com](https://istock.com/" \t "_blank)/faberfoto\_itA junior analyst at a medium-sized IT consulting firm has been asked by her manager to draft a response to an RFP from a large company that is seeking IT consulting services in connection with a new accounting system.

As the analyst worked on the RFP, she noticed a specific question about her firm’s recent experience on this type of system. To the best of her knowledge, the firm has only worked on one other accounting project in the past three years. When the manager saw the analyst’s draft response, he was upset about the way she answered the question. “You don’t have to be quite that candid,” he said. “Even though we only had one formal project, we do have several people who worked on accounting systems before they came here.”

“Yes,” the analyst replied, “But that isn’t what the question is asking.” As he left her office, the manager’s final comment was, “If we want that job, we’ll have to come up with a better answer.” Thinking about it, the analyst isn’t comfortable with anything but a straight answer. Is this an ethical question? What are her options?

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

**7.10**Summary

This chapter described system development strategies and the preparation and presentation of the system requirements document.

Traditional systems must function in various hardware and software environments, be compatible with legacy systems, and operate within the constraints of company networks and desktop computing capability. Such systems utilize Internet links and resources as enhancements. In contrast, Internet-based systems treat the web as the platform, rather than just a communication channel. Many large companies use web-based systems to handle enterprise-wide applications. Compared to traditional systems, web-based systems are more scalable, less dependent on specific hardware and software, and more adaptable to outsourcing the operation and support of a software application.

Systems analysts must consider web-based development environments such as .NET, MERN, and various outsourcing options, including ASPs and IBSs. ASPs charge subscription fees for providing application software packages. IBSs offer powerful web-based servers, software hosting, and IT support services to customers.

The web generation called Web 2.0 is fueling the expansion of information sharing, user collaboration, and social networking applications such as Twitter, LinkedIn, and Facebook. Another development, called cloud computing because of the commonly used cloud symbol for the Internet, describes an overall online software and data environment, powered by supercomputer technology that is the ultimate form of SaaS.

If a company chooses to handle its own software development needs, it can create in-house systems or purchase (and possibly customize) commercially available software packages from a software vendor or VAR.

Compared with developing an in-house system, an existing commercial software package can be an attractive alternative, because a package generally costs less, takes less time to implement, has a proven track record, and is upgraded frequently. In-house development or customizing a software package might be the best choice when a standard software package cannot meet specific business requirements or constraints. In addition to customizing software packages, companies can create user applications based on standard software that has been specially configured to enhance user productivity.

An important trend that views SaaS, rather than a product, has created new development options. SaaS is a model of software deployment in which an application is hosted as a service provided to customers over the Internet.

Offshoring, also known as *offshore outsourcing* or *global outsourcing*, refers to the practice of shifting IT development, support, and operations to other countries. In a trend similar to the outflow of manufacturing jobs over a several-decade period, many firms are sending IT work overseas. The main reason for offshoring is the same as domestic outsourcing: lower bottom-line costs. Offshore outsourcing, however, involves some unique risks and concerns.

The systems analyst’s role in the software development process depends on the specific development strategy. In-house development requires much more involvement than outsourcing or choosing a commercial package.

The most important factor in choosing a development strategy is TCO. Financial analysis tools include payback analysis, which determines how long it takes for a system to pay for itself through reduced costs and increased benefits; ROI, which compares a project’s total return with its total costs; and NPV, which analyzes the value of a project by adjusting costs and benefits to reflect the time that they occur.

The process of acquiring software involves a series of steps: Evaluate the system requirements, consider network and web-related issues, identify potential software vendors or outsourcing options, evaluate the alternatives, perform cost-benefit analysis, prepare a recommendation, and implement the solution. During software acquisition, a company can use an RFP or an RFQ. An RFP invites vendors to respond to a list of system requirements and features; an RFQ seeks bids for a specific product or service.

The system requirements document is the deliverable, or end product, of the systems analysis phase. The document details all system requirements and constraints, recommends the best solution, and provides cost and time estimates for future development work. The system requirements document is the basis for the management presentation. At this point, the firm might decide to develop an in-house system, modify the current system, purchase or customize a software package, perform additional systems analysis work, or stop all further work.

Change font size

[**help**](javascript://)

[Main content](https://ng.cengage.com/static/nbreader/ui/apps/nbreader/nbreader.html#header)

# Chapter Review

## **Key Terms**

* [**application service provider (ASP)**](javascript://)
* [**benchmark**](javascript://)
* [**build or buy**](javascript://)
* [**business process outsourcing**](javascript://)
* [**cloud computing**](javascript://)
* [**evaluation and selection team**](javascript://)
* [**evaluation model**](javascript://)
* [**fixed fee model**](javascript://)
* [**forums**](javascript://)
* [**global outsourcing**](javascript://)
* [**horizontal application**](javascript://)
* [**in-house software**](javascript://)
* [**Internet business services (IBSs)**](javascript://)
* [**logical design**](javascript://)
* [**maintenance agreement**](javascript://)
* [**make or buy**](javascript://)
* [**managed hosting**](javascript://)
* [**middleware**](javascript://)
* [**mobile devices**](javascript://)
* [**net present value (NPV)**](javascript://)
* [**offshore outsourcing**](javascript://)
* [**offshoring**](javascript://)
* [**outsourcing**](javascript://)
* [**payback analysis**](javascript://)
* [**physical design**](javascript://)
* [**request for proposal (RFP)**](javascript://)
* [**request for quotation (RFQ)**](javascript://)
* [**return on investment (ROI)**](javascript://)
* [**service desk**](javascript://)
* [**service provider**](javascript://)
* [**Software as a Service (SaaS)**](javascript://)
* [**software license**](javascript://)
* [**software package**](javascript://)
* **system requirements document**
* [**software vendors**](javascript://)
* [**subscription model**](javascript://)
* [**transaction model**](javascript://)
* [**usage model**](javascript://)
* [**user application**](javascript://)
* [**user interface**](javascript://)
* [**value-added reseller (VAR)**](javascript://)
* [**vertical application**](javascript://)
* [**Web 2.0**](javascript://)

Change font size

[**help**](javascript://)