Beyond Systems Analysis and Design

Part Four

Beyond Systems Analysis and Design

Part Four introduces you to the final phases of systems development. Chapter 18, "Systems Construction and Implementation," presents the process of constructing the system from physical design specifications and the implementation of the constructed system. It will examine the tasks involved in the development, installation, testing, and delivery of the final system into production.

18 Systems Construction and Implementation

Chapter Preview and Objectives

In this chapter you will learn more about the construction and implementation phases of systems development. These two phases construct, test, install, and deliver the final system into operation. You will know that you understand the processes of constructing and implementing a system when you can:

- Explain the purpose of the construction and implementation phases of the system's life cycle.
- Describe the system's construction and implementation phases in terms of major tasks, roles, inputs, and outputs.
- Explain several application program and system tests.
- Identify several system conversion strategies.
- Identify the chapters in this textbook that can help you actually perform the tasks of systems construction and implementation.

Although some of the techniques of systems construction and implementation are introduced in this chapter, it is *not* the intent of this chapter to teach the *techniques*. This chapter teaches only the process of construction and implementation.

Introduction

Construction has finally begun on the SoundStage Member Services system. Bob Martinez is an analyst/programmer, which means that he is expected to do some programming as well as systems analysis. Tasked with writing code to implement some of the use cases, Bob is seeing the advantage of all the analysis and design work that has gone on before. From the repository of design documents, Bob can draw essentially everything he needs to know to write his programs. His boss, Sandra, insisted that he write test scripts before he began programming. Again, the use cases told him what alternatives needed to be tested and what the results should be.

Other members of the systems analysis team are working with database programmers, application programmers, Web designers and administrators, software vendors, technical writers, and an outside firm hired to perform systems testing. They are racing to meet the deadline. But it is gratifying to see the system they designed becoming a reality.

What Is Systems Construction and Implementation?

Let's begin with definitions of systems construction and implementation. **Systems construction** is the development, installation, and testing of system components. Unfortunately, ***systems development*** is a common synonym. (We dislike that synonym since it is more frequently used to describe the ***entire*** life cycle.) **Systems implementation** is the delivery of that system into production (meaning day-to-day operation).

systems construction the development, installation, and testing of system components.

systems implementation the installation and delivery of the entire system into production.

Figure 18-1 illustrates the construction and implementation phases. Notice that the trigger for the systems construction phase is the approval of the physical design specifications resulting from the design phase. Given the design specifications, we can construct and test system components for that design. Eventually we will have built the functional system. The functional system can then be implemented or delivered as an operational system.

This chapter examines each of these phases in detail.

The Construction Phase

The purpose of the construction phase is to develop and test a functional system that fulfills business and design requirements and to implement the interfaces between the new system and existing production systems. Programming is generally recognized as a major aspect of the construction phase. But with the trend toward system solutions that involve acquiring or purchasing software packages, the implementation and integration of software components is becoming an equally, if not more, common and visible aspect of the construction phase.

In this section you will learn about several tasks involved in the construction phase of a typical systems development project. Figure 18-2 depicts the various tasks for the construction phase. Let's examine each construction phase task in greater detail.

> Task 6.1—Build and Test Networks (if Necessary)

Recall that in the requirements analysis phase of systems analysis, we established network requirements. Subsequently, during the design phase we developed distributed data and process models. Using these technical design specifications to implement the network architecture for an information system is a prerequisite for the remaining construction and implementation activities.

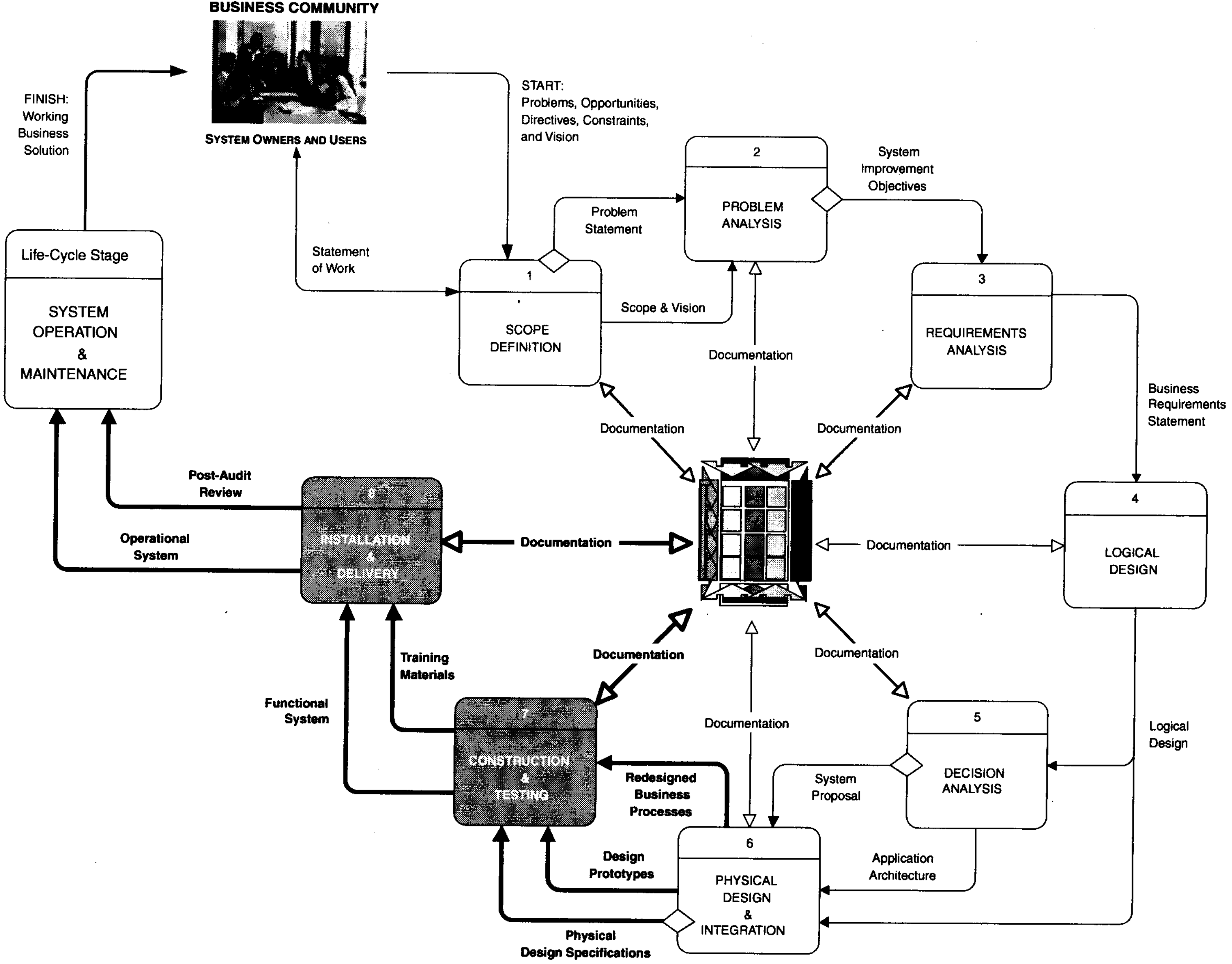


FIGURE 18-1 The Context of Systems Construction and Implementation

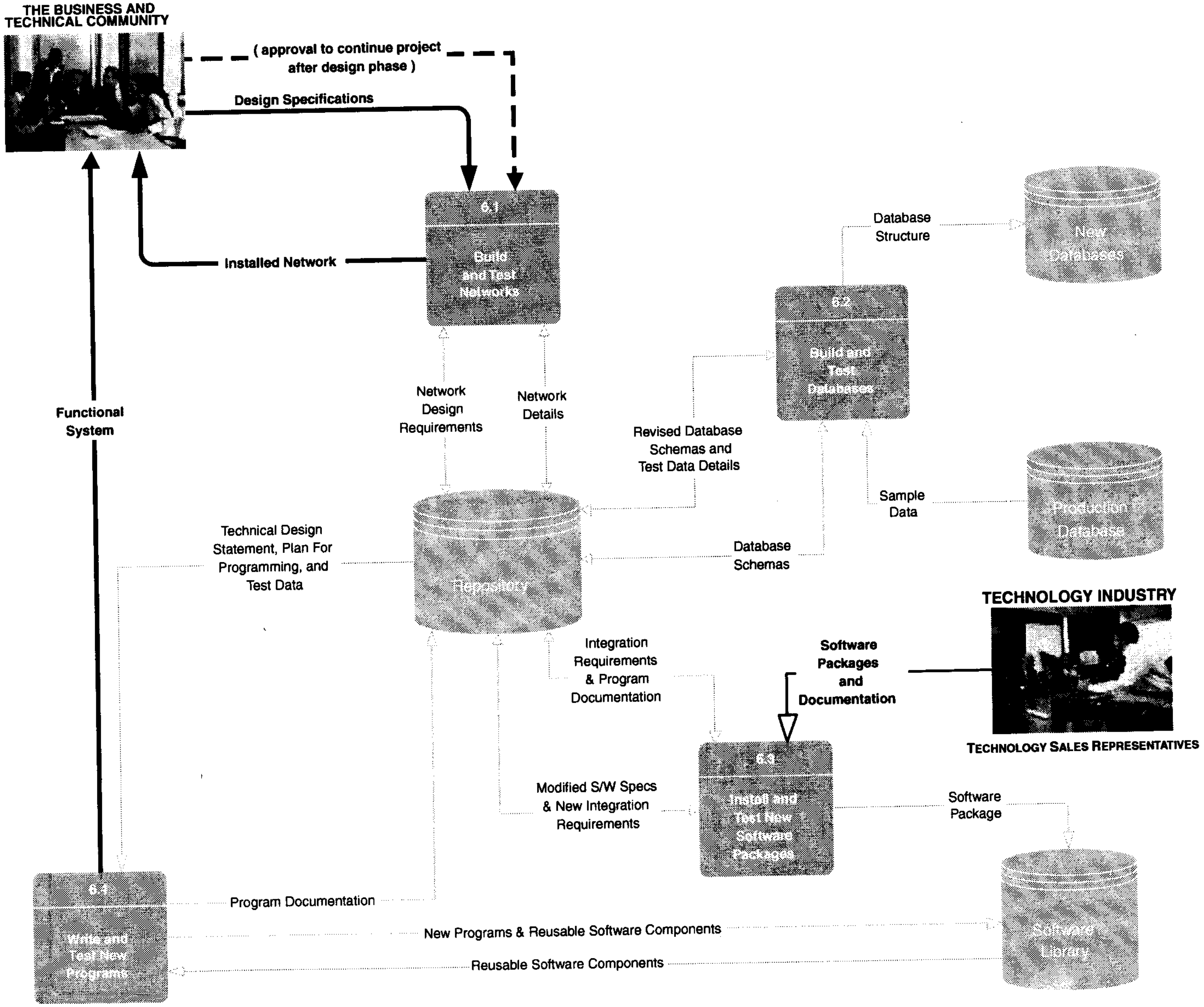

THE BUSINESS AND TECHNICAL COMMUNITY
(approval to continue project after design phase)
Design Specifications
Installed Network
Functional System
6.1
Build and Test Networks
Network Design Requirements
Network Details
Database Structure
New Databases
6.2
Build and Test Databases
Revised Database Schemas and Test Data Details
Sample Data
Production Database
Technical Design Statement, Plan For Programming, and Test Data
Repository
Database Schemas
TECHNOLOGY INDUSTRY
Integration Requirements & Program Documentation
Software Packages and Documentation
TECHNOLOGY SALES REPRESENTATIVES
6.3
Install and Test New Software Packages
Modified S/W Specs & New Integration Requirements
Software Package
6.4
Write and Test New Programs
Program Documentation
New Programs & Reusable Software Components
Reusable Software Components
Software Library

FIGURE 18-2 Systems Construction Tasks

In many cases, new or enhanced applications are built around existing networks. If so, skip this task. However, if the new application calls for new or modified networks, they must normally be implemented before building and testing databases and writing or installing computer programs that will use those networks. Thus, the first task of the construction phase may be to build and test networks.

This phase involves analysts, designers, and builders. A network designer and network administrator assume the primary responsibility for completing this task. The network designer is a specialist in the design of local and wide area networks and their connectivity. The network administrator has the expertise for building and testing network technology for the new system. He or she will also be familiar with network architecture standards that must be adhered to for any possible new networking technology. This person is also responsible for security. (The network designer and network administrator may be the same person.) While the systems analyst may be involved in the completion of this task, the analyst's role is more that of a facilitator and ensures that business requirements are not compromised by the network solution.

> Task 6.2—Build and Test Databases

Building and testing databases are unfamiliar tasks for many students, who are accustomed to having an instructor provide them with the test databases. This task must immediately precede other programming activities because databases are the resources shared by the computer programs to be written. If new or modified databases are required for the new system, we can now build and test those databases.

This task involves systems users, analysts, designers, and builders. The same system specialist that designed the databases will assume the primary responsibility in completing this task. System users may also be involved in this task by providing or approving the test data to be used in the database. When the database to be built is a noncorporate, applications-oriented database, the systems analyst often completes this task. Otherwise, systems analysts mostly ensure business requirements compliance. The database designer will often become the system builder responsible for the completion of this activity. The task may involve database programmers to build and populate the initial database and a database administrator to tune the database performance, add security controls, and provide for backup and recovery.

The primary inputs to this task are the database schema(s) specified during systems design. Sample data from production databases may be loaded into tables for testing the databases. The final product of this task is an unpopulated *database structure* for the new database. The term *unpopulated* means the database structure is implemented but data has not been loaded into the database structure. As you'll soon see, programmers will eventually write programs to populate and maintain those new databases. Revised database schema and test data details are also produced during this task and placed in the project repository for future reference.

> Task 6.3—Install and Test New Software Packages (if Necessary)

Some systems solutions may have required the purchase or lease of software packages. If so, once networks and databases for the new system have been built, we can install and test the new software. This new software will subsequently be placed in the software library.

This activity typically involves systems analysts, designers, builders, and vendors and consultants. This is the first task in the life cycle that is specific to the applications programmer. The systems analyst typically participates in the testing of the software package by clarifying requirements. Likewise, the system designer may be involved in this task to clarify integration requirements and program documentation that is to be used in testing the software. Network administrators may be involved in actually installing the software package on the network server. Finally, this task typically involves participation from the software vendor and consultants who may assist in the installation and testing process.

The main input to this task is the new software packages and documentation that are received from the system vendors. The applications programmer will complete the installation and testing of the package according to integration requirements and program documentation developed during system design. The principal deliverable of this task is the installed and tested software package that is made available in the software library. Any modified software specifications and new integration requirements that were necessary are documented and made available in the project repository to provide a history and serve as future reference.

> Task 6.4—Write and Test New Programs

We are now ready to develop (or complete) any in-house programs for the new system. Recall that prototype programs are frequently constructed in the design phase. These prototypes are included as part of the technical design specifications for completing systems construction and implementation. However, these prototypes are rarely fully functional or complete. Therefore, this activity may involve developing or refining those programs.

This task involves the systems analysts, designers, and builders. The systems analyst typically clarifies business requirements to be implemented by the programs. The designer may have to clarify the program design, integration requirements, and program documentation (developed during systems design) that is used in writing and testing the programs. The system builders will assume the primary responsibility for this activity. The applications programmer (builder) is responsible for writing and testing in-house software. Most large programming projects require a team effort. One popular organization strategy is the use of *chief programmer teams*. The team is managed by the *chief programmer*, a highly proficient and experienced programmer who assumes overall responsibility for the program design strategy, standards, and construction. The chief programmer oversees all coding and testing activities and helps with the most difficult aspects of the programs. Other team members include a *backup chief programmer*, *program librarian*, *programmers*, and *specialists*. The applications programmer is often aided by an application or software tester who specializes in building and running *test scripts* that are consistently applied to programs to test all possible events and responses.

The primary inputs to this activity are the technical design statement, plan for programming, and test data developed during systems design. Since any new programs or program components may have already been written and be in use by other existing systems, the experienced applications programmer will know to first check for possible reusable software components available in the software library. The principal deliverables of this activity are the new programs and reusable software components that are placed in the software library. This activity also results in program documentation that may need to be approved by a quality assurance group. Some information systems shops have a quality assurance group staffed by specialists who review the final program documentation for conformity to standards. This group will provide appropriate feedback regarding quality recommendations and requirements. The final program documentation is then placed in the project repository for future reference.

Testing is an important skill that is often overlooked in academic courses on computer programming. Testing should not be deferred until after the entire program has been written! There are three levels of testing to be performed: stub testing, unit or program testing, and systems testing. **Stub testing** is testing performed on individual events or modules of a program. In other words, it is the testing of an isolated subset of a program. **Unit** or **program testing** is testing in which all the events and modules that have been coded and stub tested for a program are tested as an

stub test a test performed on a subset of a program.

unit or **program test** a test performed on an entire program.

systems test a test performed on an entire system.

integrated unit; it is the testing of an entire program. **Systems testing** ensures that application programs written and tested in isolation work properly when they are integrated into the total system. A system test plan should be developed and followed for testing the system. One or more test scripts are developed for each functional and nonfunctional requirement.

Just because a single program works properly doesn't mean that it works properly with other programs. The integrated set of programs should be run through a systems test to make sure one program properly accepts, as input, the output of other programs. Once the system test is complete and determined to be successful, we can proceed to the implementation of the system.

The Implementation Phase

What's left to do? New systems usually represent a departure from the way business is currently done; therefore, the analyst must provide for a smooth transition from the old system to the new system and help users cope with normal start-up problems. Thus, the implementation phase delivers the production system into operation.

The functional system from the construction phase is the key input to the implementation phase (see Figure 18-1). The deliverable of the implementation phase (and the project) is the operational system that will enter the *operation and support* stage of the life cycle.

In your information system framework, the implementation phase considers the same building blocks as does the construction phase (see the chapter home page). In this section you will learn about several tasks involved in the implementation phase for a typical systems development project. Figure 18-3 depicts the various tasks for the implementation phase. Let's examine each implementation phase task in greater detail.

> Task 7.1—Conduct System Test

Now that the software packages and in-house programs have been installed and tested, we need to conduct a final system test. All software packages, custom-built programs, and any existing programs that comprise the new system must be tested to ensure that they all work together.

This task involves analysts, owners, users, and builders. The systems analyst facilitates the completion of this task. The systems analyst typically communicates testing problems and issues with the project team members. The system owners and system users hold the ultimate authority on whether or not a system is operating correctly. System builders, of various specialties, are involved in the systems testing. For example, applications programmers, database programmers, and networking specialists may need to resolve problems revealed during systems testing.

The primary inputs to this task include the software packages, custom-built programs, and any existing programs comprising the new system. The system test is done using the system test data that was developed earlier by the systems analyst. As with previous tests that were performed, the system test may result in required modifications to programs, thus, once again, prompting the return to a construction phase task. This iteration would continue until a successful system test was experienced.

> Task 7.2—Prepare Conversion Plan

Once a successful system test has been completed, we can begin preparations to place the new system into operation. Using the design specifications for the new system, the systems analyst will develop a detailed conversion plan. This plan will identify databases to be installed, end-user training and documentation that need to be developed, and a strategy for converting from the old system to the new system.

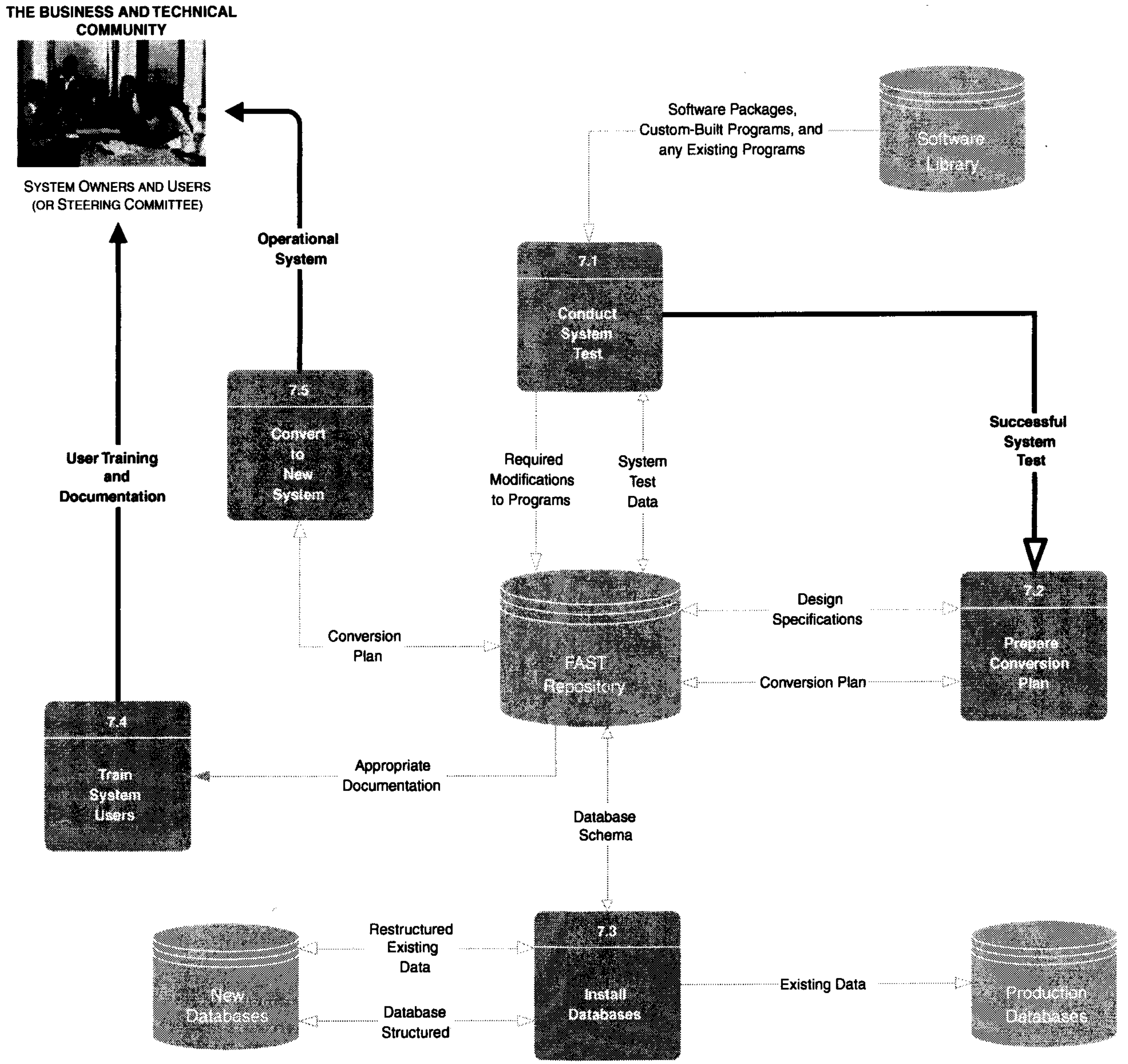


FIGURE 18-3 Systems Implementation Tasks

The project manager facilitates the activity. Systems analyst, system designer, and system builder roles are not typically involved unless deemed necessary by the project manager. Finally, many organizations require that all project plans be formally presented to a steering body (sometimes called a *steering committee*) for final approval.

This activity is triggered by the completion of a successful system test. Using the design specifications for the new system, a detailed conversion plan can be assembled. The principal deliverable of this activity is the conversion plan.

The conversion plan may include one of the following commonly used installation strategies:

- *Abrupt cut-over*—On a specific date (usually a date that coincides with an official business period such as month, quarter, or fiscal year), the old system is terminated and the new system is placed into operation. This is a high-risk approach because there may still be major problems that won't be uncovered until the system has been in operation for at least one business period. On the other hand, there are no transition costs. Abrupt cut-over may be necessary if, for instance, a government mandate or business policy becomes effective on a specific date and the system couldn't be implemented before that date.
- *Parallel conversion*—Under this approach, both the old and the new systems are operated for some time period. This ensures that all major problems in the new system have been solved before the old system is discarded. The final cut-over may be either abrupt (usually at the end of one business period) or gradual, as portions of the new system are deemed adequate. This strategy minimizes the risk of major flaws in the new system causing irreparable harm to the business; however, it also means the cost of running two systems over some period must be incurred. Because running two editions of the same system on the computer could place an unreasonable demand on computing resources, this may be possible only if the old system is largely manual.
- *Location conversion*—When the same system will be used in numerous geographical locations, it is usually converted at one location first (using either abrupt or parallel conversion). As soon as that site has approved the system, it can be farmed to the other sites. Other sites can be cut over abruptly because major errors have been fixed. Furthermore, other sites benefit from the learning experiences of the first test site. The first production test site is often called a ***beta test site.***
- *Staged conversion*—Like location conversion, staged conversion is a variation on the abrupt and parallel conversions. A staged conversion is based on the version concept introduced earlier. Each successive version of the new system is converted as it is developed. Each version may be converted using the abrupt, parallel, or location strategy.

The conversion plan also typically includes a systems acceptance test plan. The systems acceptance test is the final opportunity for end users, management, and information systems operations management to accept or reject the system. A **systems acceptance test** is a final system test performed by end users using real data over an extended time period. It is an extensive test that addresses three levels of acceptance testing—verification testing, validation testing, and audit testing:

systems acceptance test a test performed on the final system wherein users conduct verification, validation, and audit tests.

- *Verification testing* runs the system in a simulated environment using simulated data. This simulated test is sometimes called *alpha testing.* The simulated test is primarily looking for errors and omissions regarding end-user and design specifications that were specified in the earlier phases but not fulfilled during construction.
- *Validation testing* runs the system in a live environment using real data. This is sometimes called *beta testing*. During this validation, a number of items are tested:

  a. *Systems performance.* Is the throughput and response time for processing adequate to meet a normal processing workload? If not, some programs may have to be rewritten to improve efficiency, or processing hardware may have to be replaced or upgraded to handle the additional workload.

b. *Peak workload processing performance.* Can the system handle the workload during peak processing periods? If not, improved hardware and/or software may be needed to increase efficiency, or processing may need to be rescheduled—that is, consider doing some of the less critical processing during nonpeak periods.

c. *Human engineering test.* Is the system as easy to learn and use as anticipated? If not, is it adequate? Can enhancements to human engineering be deferred until after the system has been placed into operation?

d. *Methods and procedures test.* During conversion, the methods and procedures for the new system will be put to their first real test. Methods and procedures may have to be modified if they prove to be awkward and inefficient from the end users' standpoint.

e. *Backup and recovery testing.* All backup and recovery procedures should be tested. This should include simulating a data loss disaster and testing the time required to recover from that disaster. Also, a before-and-after comparison of the data should be performed to ensure that data was properly recovered. It is crucial to test these procedures. Don't wait until the first disaster to find an error in the recovery procedures.

- **Audit testing** certifies that the system is free of errors and is ready to be placed into operation. Not all organizations require an audit. But many firms have an independent audit or quality assurance staff that must certify a system's acceptability and documentation before that system is placed into final operation. There are independent companies that perform systems and software certification for end users' organizations.

audit test a test performed to ensure a new system is ready to be placed into operation.

> Task 7.3—Install Databases

Recall that in a previous phase you built and tested databases.To place the system into operation, you will need fully loaded (or "populated") databases.Therefore, the next task we'll survey is installation of databases.The purpose of this task is to populate the new system's databases with existing data from the old system.

At first, this activity may seem trivial. But consider the implications of loading a typical table, say, MEMBER.Tens or hundreds of thousands of records may have to be loaded. Each must be input, edited, and confirmed before the database table is ready to be placed into operation.

Systems builders play a primary role in this activity.The task will normally be completed by application programmers who will write the special programs to extract data from existing databases and programs to populate the new databases. Systems analysts and designers may play a small role in completing this activity.Their primary involvement will be the calculating of database sizes and estimating of the time required to perform the installation. Finally, data entry personnel or hired help may often be assigned to do data entry.

Special programs will have to be written to populate the new databases. Existing data from the production databases, coupled with the database schema(s) models and database structures for the new databases, will be used to write computer programs to populate the new databases with restructured existing data.The principal deliverable of this task is the restructured existing data that has been populated in the databases for the new system.

> Task 7.4—Train Users

Change may be good, but it's not always easy. Converting to a new system necessitates that system users be trained and provided with documentation (user manuals) that guides them through using the new system.

Training can be performed one on one; however, group training is generally preferred. It is a better use of your time, and it encourages group-learning possibilities. Think about your education for a moment. You really learn more from your fellow students and colleagues than from your instructors. Instructors facilitate learning and instruction, but you master specific skills through practice with large groups where common problems and issues can be addressed more effectively. Take advantage of the ripple effect of education. The first group of trainees can then train several other groups.

The task is completed by the systems analyst and involves system owners and users. Given appropriate documentation for the new system, the systems analysts will provide end-user documentation (typically in the form of manuals) and training for the system users. The system owners must support this activity. They must be willing to approve the release time necessary for people to obtain the training needed to become successful users of the new system. Remember, the system is for the user! User involvement is also important in this activity because the end users will inherit the successes and failures from this effort. Fortunately, users' involvement during this task is rarely overlooked. The most important aspect of their involvement is training and advising the users. They must be trained to use equipment and to follow the procedures required of the new system. But no matter how good the training is, users will become confused at times. Or perhaps they will find mistakes or limitations. Thus, it is the responsibility of the analyst to help the users through the learning period until they become comfortable with the new system.

Given appropriate documentation for the new system, the systems analyst will provide the system users with the documentation and training needed to properly use the new system. The principal deliverable of this task is user training and documentation. Many organizations hire special systems analysts who do nothing but write user documentation and training guides. If you have a skill for writing clearly, the demand for your services is out there! Figure 18-4 is a typical outline for a training manual. The Golden Rule should apply to user manual writing: "Write unto others as you would have them write unto you." You are not a business expert. Don't expect the reader to be a technical expert. Every possible situation and its proper procedure must be documented.

>Task 7.5—Convert to New System

Conversion to the new system from the old system is a significant milestone. After conversion, the ownership of the system officially transfers from the analysts and programmers to the end users. The analyst completes this task by carefully carrying out the conversion plan. Recall that the conversion plan includes detailed installation

FIGURE 18-4 An Outline for a Training Manual

Training Manual End Users Guide Outline

I. Introduction.
II. Manual.
   A. The manual system (a detailed explanation of people's jobs and standard operating procedures for the new system).
   B. The computer system (how it fits into the overall work flow).
      1. Terminal/keyboard familiarization.
      2. First-time end users.
         a. Getting started.
         b. Lessons.
   C. Reference manual (for nonbeginners).
III. Appendixes.
   A. Error messages.

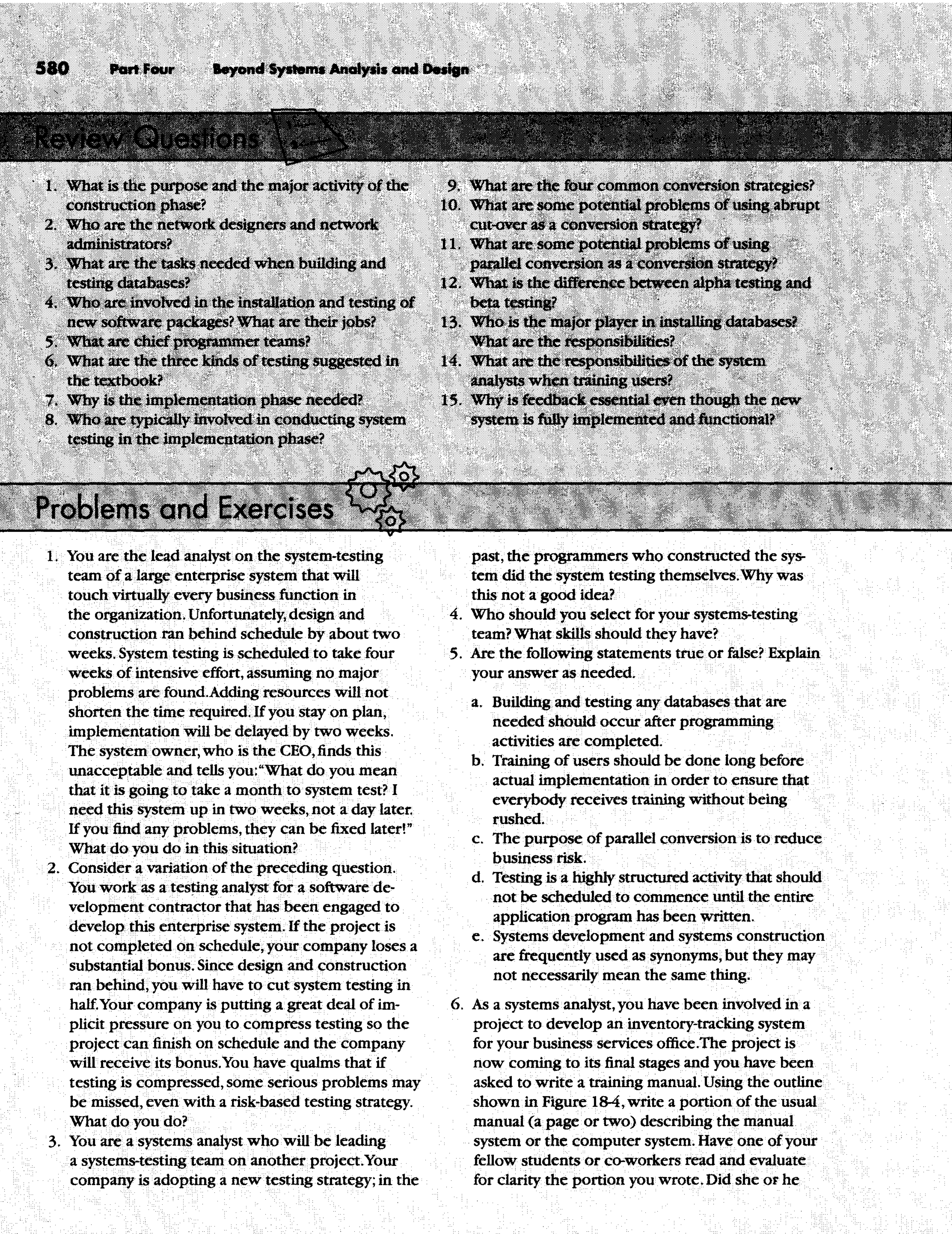
strategies to follow for converting from the existing to the new production information system. This task also involves completing a systems audit.

The task involves the systems owners, users, analysts, designers, and builders. The project manager who will oversee the conversion process facilitates it. The system owners provide feedback regarding their experiences with the overall project. They may also provide feedback regarding the new system that has been placed into operation. The system users will provide valuable feedback pertaining to the actual use of the new system. They will be the source of the majority of the feedback used to measure the system's acceptance. The systems analysts, designers, and builders will assess the feedback received from the system owners and users once the system is in operation. In many cases, that feedback may stimulate actions to correct identified shortcomings. Regardless, the feedback will be used to help benchmark new systems projects down the road.

The key input to this activity is the conversion plan that was created in an earlier implementation phase task. The principal deliverable is the operational system that is placed into production in the business.

Chapter Review

1. Systems construction is the development, installation, and testing of system components.
2. Systems implementation is the delivery of the system into production (meaning day-to-day operation).
3. The purpose of the construction phase is to develop and test a functional system that fulfills business and design requirements and to implement the interfaces between the new system and existing production systems.
4. The construction phase consists of four tasks: build and test networks, build and test databases, install and test new software packages, and write and test new programs.
5. Three levels of testing are performed on new programs:
   a. Stub testing is testing performed on individual modules, whether they be main program, subroutine, subprogram, block, or paragraph.
   b. Unit or program testing is testing in which all the modules that have been coded and stub tested are tested as an integrated unit.
   c. Systems testing ensures that application programs written in isolation work properly when they are integrated into the total system.
6. The purpose of the implementation phase is to smoothly convert from the old system to the new system.
7. The systems implementation consists of the following activities: conducting a system test, preparing a systems conversion plan, installing databases, training system users, and converting from the old system to the new system.
8. There are several commonly used strategies for converting from an existing to a new production information system, including:
   a. Abrupt cut-over—On a specific date, the old system is terminated and the new system is placed into operation.
   b. Parallel conversion—Both the old and the new systems are operated for some time period to ensure that all major problems in the new system are solved before the old system is discarded.
   c. Location conversion—When the same system will be used in numerous geographical locations, it is usually converted at one location and, following approval, farmed to the other sites.
   d. Staged conversion—Each successive version of the new system is converted as it is developed. Each version may be converted using the abrupt, parallel, or location strategies.
9. The systems acceptance test is the final opportunity for end users, management, and information systems operations management to accept or reject the system. A systems acceptance test is a final system test performed by end users using real data over an extended period. It is an extensive test that addresses three levels of acceptance testing—verification testing, validation testing, and audit testing:
   a. Verification testing runs the system in a simulated environment using simulated data.
   b. Validation testing runs the system in a live environment using real data. This is sometimes called beta testing.
   c. Audit testing certifies that the system is free of errors and is ready to be placed into operation.



Review Questions

1. What is the purpose and the major activity of the construction phase?
2. Who are the network designers and network administrators?
3. What are the tasks needed when building and testing databases?
4. Who are involved in the installation and testing of new software packages? What are their jobs?
5. What are chief programmer teams?
6. What are the three kinds of testing suggested in the textbook?
7. Why is the implementation phase needed?
8. Who are typically involved in conducting system testing in the implementation phase?
9. What are the four common conversion strategies?
10. What are some potential problems of using abrupt cut-over as a conversion strategy?
11. What are some potential problems of using parallel conversion as a conversion strategy?
12. What is the difference between alpha testing and beta testing?
13. Who is the major player in installing databases? What are the responsibilities?
14. What are the responsibilities of the system analysts when training users?
15. Why is feedback essential even though the new system is fully implemented and functional?

Problems and Exercises

1. You are the lead analyst on the system-testing team of a large enterprise system that will touch virtually every business function in the organization. Unfortunately, design and construction ran behind schedule by about two weeks. System testing is scheduled to take four weeks of intensive effort, assuming no major problems are found. Adding resources will not shorten the time required. If you stay on plan, implementation will be delayed by two weeks. The system owner, who is the CEO, finds this unacceptable and tells you: "What do you mean that it is going to take a month to system test? I need this system up in two weeks, not a day later. If you find any problems, they can be fixed later!" What do you do in this situation?
2. Consider a variation of the preceding question. You work as a testing analyst for a software development contractor that has been engaged to develop this enterprise system. If the project is not completed on schedule, your company loses a substantial bonus. Since design and construction ran behind, you will have to cut system testing in half. Your company is putting a great deal of implicit pressure on you to compress testing so the project can finish on schedule and the company will receive its bonus. You have qualms that if testing is compressed, some serious problems may be missed, even with a risk-based testing strategy. What do you do?
3. You are a systems analyst who will be leading a systems-testing team on another project. Your company is adopting a new testing strategy; in the past, the programmers who constructed the system did the system testing themselves. Why was this not a good idea?
4. Who should you select for your systems-testing team? What skills should they have?
5. Are the following statements true or false? Explain your answer as needed.
   a. Building and testing any databases that are needed should occur after programming activities are completed.
   b. Training of users should be done long before actual implementation in order to ensure that everybody receives training without being rushed.
   c. The purpose of parallel conversion is to reduce business risk.
   d. Testing is a highly structured activity that should not be scheduled to commence until the entire application program has been written.
   e. Systems development and systems construction are frequently used as synonyms, but they may not necessarily mean the same thing.
6. As a systems analyst, you have been involved in a project to develop an inventory-tracking system for your business services office. The project is now coming to its final stages and you have been asked to write a training manual. Using the outline shown in Figure 18-4, write a portion of the usual manual (a page or two) describing the manual system or the computer system. Have one of your fellow students or co-workers read and evaluate for clarity the portion you wrote. Did she or he

find it understandable and clear? Did it provide the appropriate level of detail that an end user would need?

7. During systems construction and implementation, aren't most of the activities technical in nature, so that users don't need to be involved except for system testing?

8. Match the terms in the first column with the definitions or examples in the second column:

1. Beta testing	A. Production database without data loaded in
2. Alpha testing	B. Testing of throughput/ response time under normal load
3. Program testing	C. Migrating completed system into production environment
4. Audit testing	D. Unanticipated sudden system shutdown testing
5. System performance testing	E. Application program-level code testing
6. Unpopulated database	F. Independently performed certification-level testing
7. Backup and recovery testing	G. Module-level testing of code
8. Peak performance testing	H. Extensive verification, validation, and audit testing
9. Abrupt cut-off	I. Environmental-level testing of application program(s)
10. System implementation	J. Environmental-level testing by users with simulated data
11. Systems acceptance testing	K. Live environmental-level testing by users with live data
12. System testing	L. Testing of throughput/response time under load spikes
13. Stub testing	M. Installation strategy type

9. "The goal of human interface design is to create a system that is intuitive to use. To require a users manual is an admission of failure." Respond to this statement. Do you agree or disagree with it? Explain why.

10. Many organizations require a postimplementation evaluation report (PIER), usually somewhere between six months and a year after implementation. What purpose(s) does this serve?

11. If a project is poorly designed and constructed, will a well-planned and well-executed implementation effort help the project to succeed? What about the opposite situation? Will a well-designed and well-constructed system overcome a poor implementation effort?

Projects and Research

1. A number of companies, such as Mercury Interactive of Rational (now owned by IBM), offer automated software-testing packages as stand-alone products or as an integrated part of a larger suite. Research these software packages on the Web and trade journals. Download and try any trial versions you find. In addition, contact the software-testing staff in several local organizations and interview them regarding their software-testing methods.

   a. Describe your research—what products did you find?

   b. Compare and contrast their features and functionality.

   c. Did the software testers you contacted use an automated software testing tool? If so, was it a home-grown tool or a commercial product? Did they indicate any preference as to which one they thought was best?

   d. If you were the testing manager and were given the choice to purchase any automated software-testing tool, which one would you select? Or would you prefer to develop your own homegrown automated testing program? Explain your answer.

   e. What do you see as the primary advantages to using an automated software-testing package?

2. You are a systems analyst working on a major project for an organization that has several hundred employees in its headquarters office and about a dozen offices located in this country, in Canada, and in Mexico. The objective of the project is to implement an enterprise-wide mission-critical information system. The project is now in the construction and implementation phase, and you have been assigned responsibility for selecting the conversion strategy and for developing the

conversion plan. Prepare a summary analysis of the different installation strategies discussed in the book, and recommend the one that you feel would be most appropriate.

3. Assuming that your recommendation is accepted by management, draft a detailed conversion plan that addresses the actual implementation strategy for converting to the new system. After you complete the draft plan, have it reviewed by one or more IT staff members with experience in developing conversion plans. Make any needed changes and repeat the review process until the consensus is that your plan is realistic and doable.
4. The next step is to prepare the systems acceptance test plan. Using the material in the textbook as a general guideline, research on the Web some of the more detailed components that go into acceptance testing. Select some of the testing templates that should be readily available, and modify them as appropriate. Then draft the test plan and share it with IT testing staff. Make sure that your plan addresses any potential risks. Make any needed changes and repeat the review process as necessary with the testing staff until your plan is ready to put into action.
5. The textbook describes a traditional method for delivering end-user training. Are there other methods, which are Web-based, that may offer more effective and/or efficient methods for delivering end-user training? Research some of the Web-based, training methods that are becoming more widely used. Then use the scenario described in Question 2 to develop a Web-based end-user training plan. After you draft the plan, have some professional trainers review it for completeness and feasibility and make any needed changes. Then have some of your fellow students or co-workers review your training plan from the perspective of an end-user. Were you able to develop a feasible plan?
6. There is an unwritten principle that says that no matter how much you plan for a system implementation, something unanticipated will almost always happen, often at the worst possible moment. Interview several analysts in local organizations who have expertise in implementing systems. Ask them about their experiences, what their worst horror story was, and what they learned from it. Supplement these interviews with research on contingency implementation planning. Then use your anecdotal information and research to put together a set of guidelines on planning for and handling the unexpected during system implementation.

Minicases

1. Wow Munchies has a Web site, *www.wowmunchies.com*, which is currently hosted on server 123coolhost at a Web-hosting company called Cool Hosting. But Wow Munchies has decided to have its Web pages updated and serviced by another hosting company, Reliable Host, using the server 123reliable. The new hosting company pointed the DNS for *www.wowmunchies.com* to server 123reliable *before* it had the Web pages loaded and tested on its server. It takes 12–72 hours for the DNS change to take place, and Reliable thought it would have the Web pages up in the lag time. It wasn't able to. As a result, the DNS pointed to the new server for several days before the new site was functioning again. Wow Munchies lost an estimated $200,000 in revenues as a result of the sites downtime. Comment on what went wrong, and how it could have been avoided.

Team and Individual Exercises

1. Individual: Present one of the deliverables from this class (any deliverable) to the class. Dress professionally, and use an interesting presentation technology. Remember to dress professionally, speak slowly and clearly, and have fun. No one knows your deliverable better than you!
2. Professor/class: Weather permitting, have class outside.
3. Individual/class: Create a networking sheet of contact information for everyone in the class (and other students if they wish to join). Everyone who submits information gets a copy. Keep in touch after you leave school!

Suggested Readings

Bell, P., and C. Evans. *Mastering Documentation.* New York: John Wiley & Sons, 1989.

Boehm, Barry. "Software Engineering." *IEEE Transactions on Computers, C-25,* December 1976. This classic paper demonstrated the importance of catching errors and omissions before programming begins.

Brooks, F. P. *The Mythical Man-Month.* Reading, MA: Addison-Wesley, 1995.

Metzger, Philip W. *Managing a Programming Project,* 2nd ed. Englewood Cliffs, NJ: Prentice Hall, 1981. This is one of the few books to place emphasis solely on systems implementation.

Mosely, D. J. *The Handbook of MIS Application Software Testing.* Englewood Cliffs, NJ: Yourdon Press, 1993.

Back Matter