19

Systems Construction and Implementation

Overview

Chapter 19 teaches students more about the systems construction and implementation phases of systems development. 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 only teaches the process of systems construction and implementation.

Chapter to Course Sequencing

Students are encouraged to read Chapter 3 to provide perspective for systems construction and implementation. It would also be beneficial if Chapters 5 and 12 were read before this chapter. This would provide a better perspective and foundation for studying Chapter 19.

What's Different Here and Why?

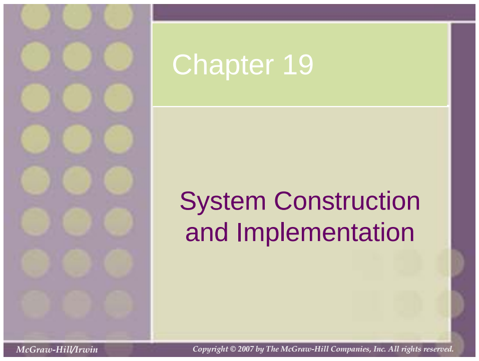
This chapter did not necessitate many changes from the sixth edition.

1. As with all chapters, we have streamlined the SoundStage episode into a quick narrative introduction to the concepts presented the chapter.
2. We updated all technology references throughout the chapter.

Lesson Planning Notes for Slides

The following instructor notes, keyed to slide images from the PowerPoint repository, are intended to help instructors integrate the slides into their individual lesson plans for this chapter.

Slide 1



Chapter 19

System Construction and Implementation

McGraw-Hill/Irwin

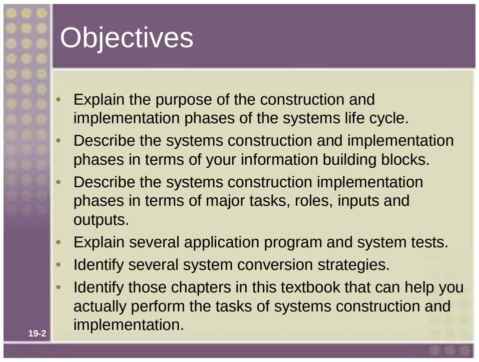
Copyright © 2007 by The McGraw-Hill Companies, Inc. All rights reserved.

slide appearance after initial mouse click in slide show mode

This repository of slides is intended to support the named chapter. The slide repository should be used as follows:
Copy the file to a unique name for your course and unit.
Edit the file by deleting those slides you don't want to cover, editing other slides as appropriate to your course, and adding slides as desired.
Print the slides to produce transparency masters or print directly to film or present the slides using a computer image projector.

Each slide includes instructor notes. To view those notes in PowerPoint, click-left on the View Menu; then click left on Notes View sub-menu. You may need to scroll down to see the instructor notes.
The instructor notes are also available in hard-copy as the Instructor Guide to Accompany Systems Analysis and Design Methods, 6/ed.

Slide 2



Objectives

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

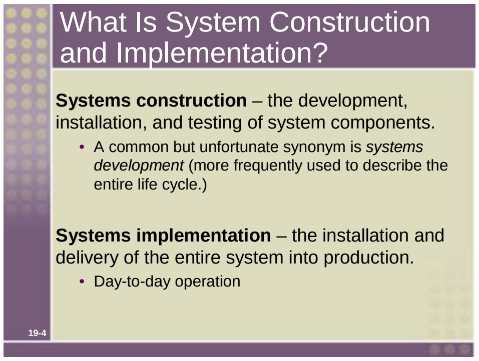
19-2

No additional notes.

Slide 3

Teaching Notes
This slide shows the how this chapter's content fits with the building blocks framework used throughout the textbook. The emphasis of this chapter is with the construction & testing phase and installation & delivery phase, spanning the communication focus, knowledge focus, and process focus. It involves system builders and systems analysts, and users.

Slide 4



What Is System Construction and Implementation?

Systems construction – the development, installation, and testing of system components.

- A common but unfortunate synonym is *systems development* (more frequently used to describe the entire life cycle.)

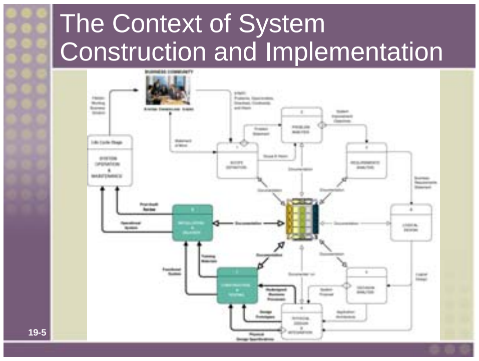
Systems implementation – the installation and delivery of the entire system into production.

- Day-to-day operation

19-4

Teaching Notes
Depending on the development techniques used, portions of these phases may have been completed. For example, prototyping may have resulted in the construction of system components as well as the testing and training.

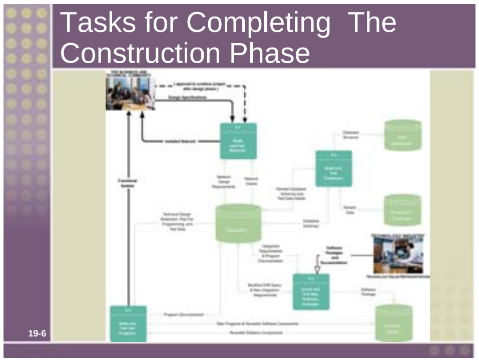
Slide 5



The Context of System Construction and Implementation

No additional notes.

Slide 6

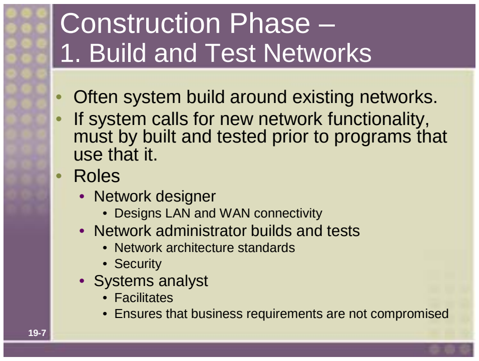


Tasks for Completing The Construction Phase

19-6

No additional notes.

Slide 7



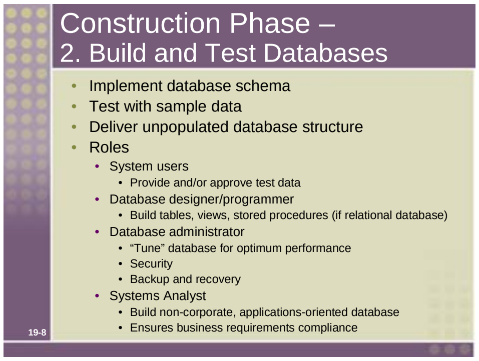
Construction Phase – 1. Build and Test Networks

- Often system build around existing networks.
- If system calls for new network functionality, must by built and tested prior to programs that use that it.
- Roles
  - Network designer
    - Designs LAN and WAN connectivity
  - Network administrator builds and tests
    - Network architecture standards
    - Security
  - Systems analyst
    - Facilitates
    - Ensures that business requirements are not compromised

19-7

No additional notes.

Slide 8



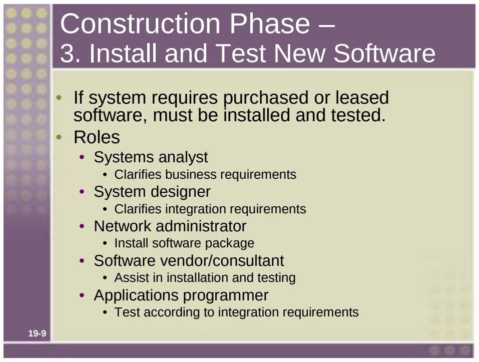
Construction Phase – 2. Build and Test Databases

- Implement database schema
- Test with sample data
- Deliver unpopulated database structure
- Roles
  - System users
    - Provide and/or approve test data
  - Database designer/programmer
    - Build tables, views, stored procedures (if relational database)
  - Database administrator
    - "Tune" database for optimum performance
    - Security
    - Backup and recovery
  - Systems Analyst
    - Build non-corporate, applications-oriented database
    - Ensures business requirements compliance

19-8

No additional notes.

Slide 9



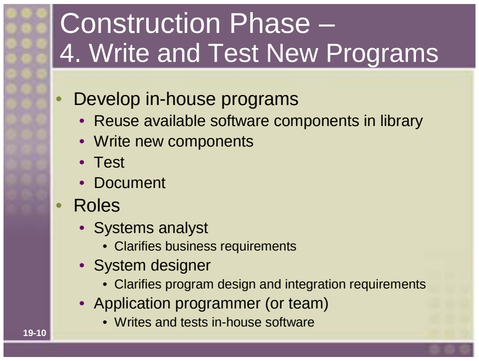
Construction Phase – 3. Install and Test New Software

- If system requires purchased or leased software, must be installed and tested.
- Roles
  - Systems analyst
    - Clarifies business requirements
  - System designer
    - Clarifies integration requirements
  - Network administrator
    - Install software package
  - Software vendor/consultant
    - Assist in installation and testing
  - Applications programmer
    - Test according to integration requirements

19-9

No additional notes.

Slide 10



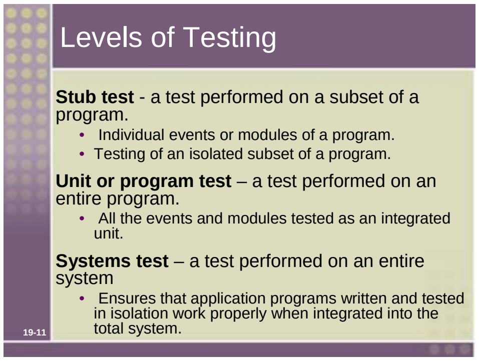
Construction Phase – 4. Write and Test New Programs

- Develop in-house programs
  - Reuse available software components in library
  - Write new components
  - Test
  - Document
- Roles
  - Systems analyst
    - Clarifies business requirements
  - System designer
    - Clarifies program design and integration requirements
  - Application programmer (or team)
    - Writes and tests in-house software

19-10

No additional notes.

Slide 11



Levels of Testing

Stub test - a test performed on a subset of a program.

- Individual events or modules of a program.
- Testing of an isolated subset of a program.

Unit or program test – a test performed on an entire program.

- All the events and modules tested as an integrated unit.

Systems test – a test performed on an entire system

- Ensures that application programs written and tested in isolation work properly when integrated into the total system.

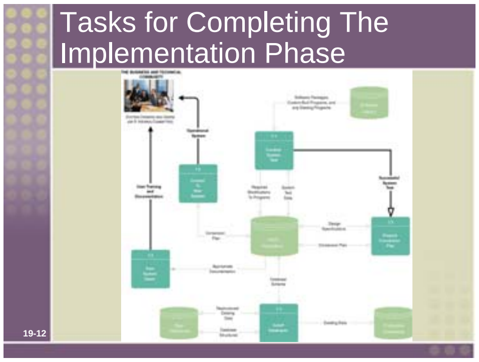
19-11

Teaching Notes

Describe several scenarios of testing and ask the students to characterize them as stub, unit, or system level testing.

Share with the class personal case studies of the ramifications of failure to properly test.

Slide 12



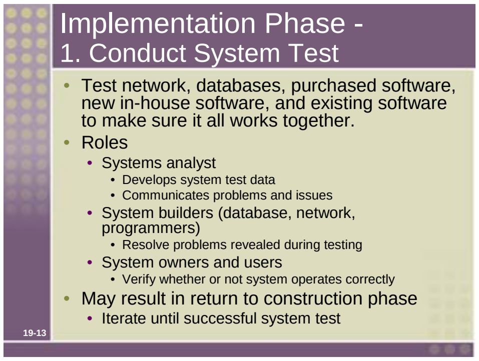
Tasks for Completing The Implementation Phase

THE BUSINESS AND TECHNICAL COMMUNITY
Operational System
User Training and Documentation
Software Packages, Custom-Built Programs, and any Existing Programs
Required Modifications To Programs
System Test Data
Successful System Test
Design Specifications
Conversion Plan
Conversion Plan
Appropriate Documentation
Database Schema
Restructured Existing Data
Database Structures
Existing Data

19-12

No additional notes

Slide 13



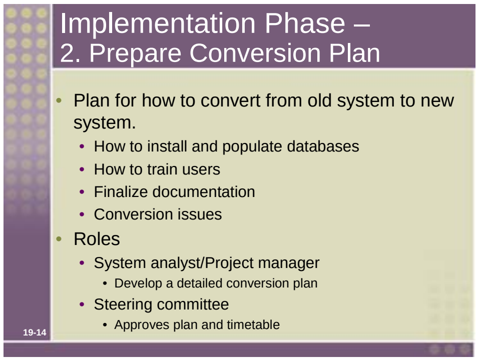
Implementation Phase - 1. Conduct System Test

- Test network, databases, purchased software, new in-house software, and existing software to make sure it all works together.
- Roles
  - Systems analyst
    - Develops system test data
    - Communicates problems and issues
  - System builders (database, network, programmers)
    - Resolve problems revealed during testing
  - System owners and users
    - Verify whether or not system operates correctly
- May result in return to construction phase
  - Iterate until successful system test

19-13

No additional notes

Slide 14



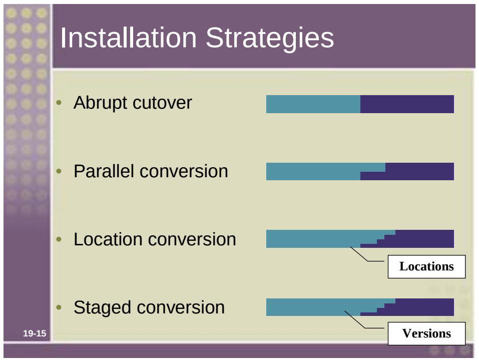
Implementation Phase – 2. Prepare Conversion Plan

- Plan for how to convert from old system to new system.
  - How to install and populate databases
  - How to train users
  - Finalize documentation
  - Conversion issues
- Roles
  - System analyst/Project manager
    - Develop a detailed conversion plan
  - Steering committee
    - Approves plan and timetable

19-14

No additional notes

Slide 15



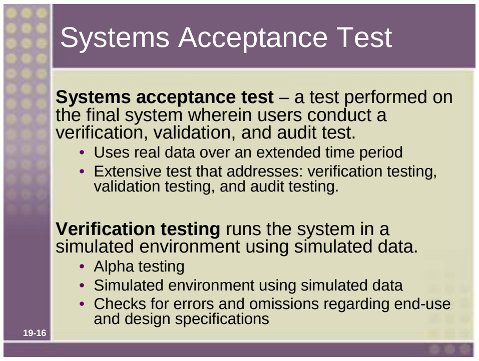
Installation Strategies

- Abrupt cutover
- Parallel conversion
- Location conversion
- Staged conversion

19-15

Teaching Notes
Discuss the risks and potential rewards of each strategy.

Slide 16



Systems Acceptance Test

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

- Uses real data over an extended time period
- Extensive test that addresses: verification testing, validation testing, and audit testing.

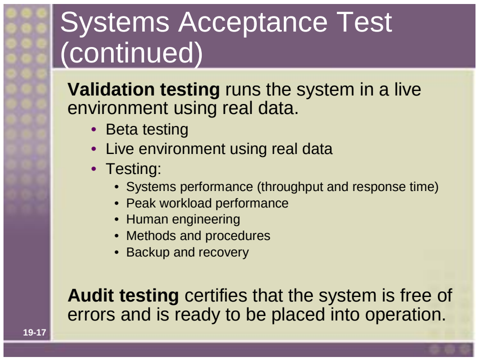
Verification testing runs the system in a simulated environment using simulated data.

- Alpha testing
- Simulated environment using simulated data
- Checks for errors and omissions regarding end-use and design specifications

19-16

No additional notes

Slide 17



Systems Acceptance Test (continued)

Validation testing runs the system in a live environment using real data.

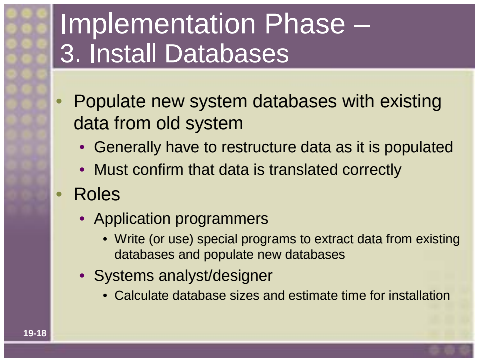
- Beta testing
- Live environment using real data
- Testing:
  - Systems performance (throughput and response time)
  - Peak workload performance
  - Human engineering
  - Methods and procedures
  - Backup and recovery

Audit testing certifies that the system is free of errors and is ready to be placed into operation.

19-17

No additional notes.

Slide 18



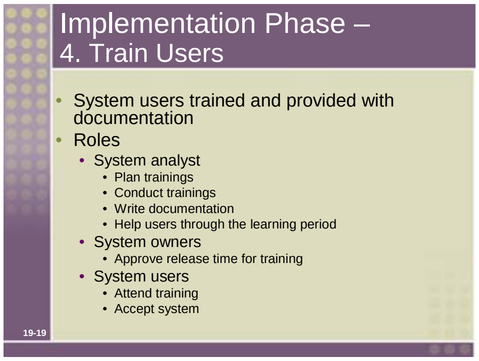
Implementation Phase – 3. Install Databases

- Populate new system databases with existing data from old system
  - Generally have to restructure data as it is populated
  - Must confirm that data is translated correctly
- Roles
  - Application programmers
    - Write (or use) special programs to extract data from existing databases and populate new databases
  - Systems analyst/designer
    - Calculate database sizes and estimate time for installation

19-18

No additional notes

Slide 19



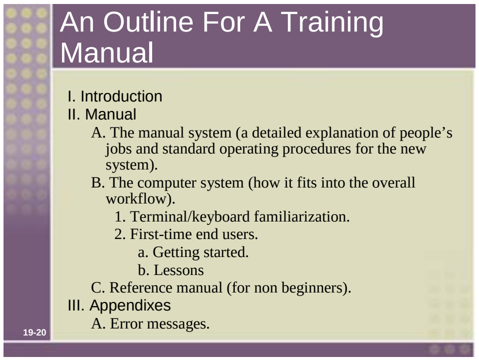
Implementation Phase – 4. Train Users

- System users trained and provided with documentation
- Roles
  - System analyst
    - Plan trainings
    - Conduct trainings
    - Write documentation
    - Help users through the learning period
  - System owners
    - Approve release time for training
  - System users
    - Attend training
    - Accept system

19-19

No additional notes

Slide 20



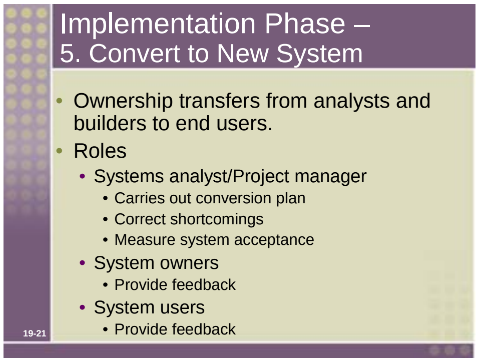
An Outline For A Training Manual

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 workflow).
1. Terminal/keyboard familiarization.
2. First-time end users.
a. Getting started.
b. Lessons
C. Reference manual (for non beginners).
III. Appendixes
A. Error messages.

19-20

No additional notes.

Slide 21



Implementation Phase – 5. Convert to New System

- Ownership transfers from analysts and builders to end users.
- Roles
  - Systems analyst/Project manager
    - Carries out conversion plan
    - Correct shortcomings
    - Measure system acceptance
  - System owners
    - Provide feedback
  - System users
    - Provide feedback

19-21

No additional notes

Answers to End of Chapter Questions and Exercises

Review Questions

1. The purpose of the construction phase is to develop a functional system which will fulfill business and design requirements. After development, testing is also needed. The major activity of this phase is programming, although acquiring software packages is coming popular.

2. Network designers are responsible for designing the local and wide area network (LAN and WAN) and how they connect with each other. The network administrators are in charge of building and testing the network. They also need to specify the network architecture standards and maintain network security.

3. First, database schemas that are specified in the system design phase are needed as a major input. After that, we can load the data from the production databases into the tables of the database to conduct any testing activities. Lastly, the product is an unpopulated database structure, meaning a database being implemented without actual data loaded into it. Programmers will be in charge of populating the database through writing programs.

4. - System analysts will need to make sure the requirements are clear during this process.
   - System designers will also need to make sure the integration requirements and program documentation are clear.
   - Network administrators will need to install the software package on the server.
   - Software vendors and consultants will provide help in the process if necessary.

5. Chief programmer teams are one common strategy used by organization when it comes to programming. This team is supervised by a very experienced and proficient programmer or chief programmer, who is responsible for the overall program design strategy, standards, and constructions. The chief programmer also manages all the activities regarding coding and testing. In addition to the chief programmer, there are other members in this programmer team including backup chief programmer, program librarian, programmers, and specialists.

6. Stub test testing done on the modules of a program.

Unit or program test: testing done after the stub test for a program. This kind of test tests all the events and modules together.

System test: All the programs written and tested will be tested here as one integrated system to make sure everything works together properly.

7. The implementation phase is needed because in order for a new system to work effectively in an organization, a smooth transition from the old to the new system is essential. It is also necessary to provide the system users with assistance during this transition period. The goal of the implementation phase is to make the production system into operation.

8.
   - System analysts will talk to the project team members and let them know if there are any problems or issues during testing.
   - System owners and users will decide if a system is working properly or not.
   - System builders, such as programmers, database programmers, and networking specialists, will help solve problems identified in the testing.

9.
   - Abrupt cut-over
     The old system, under this strategy, will be shut down on a specific day, and the new system will replace the old one and will be used for operation. This approach bears a very high risk because the new system may not be functioning as anticipated and problems may arise during the transition period.

   - Parallel conversion
     Using this strategy, the old and the new systems will be in use for a period of time together. This approach has less risk because the old systems can serve as a back up while the new system is put in operation.

   - Location conversion
     Since some systems may need to be used in different geographic location, location conversion will make sure the system is working properly in one location before implementing the systems in the other locations.

   - Staged conversion
     Systems are implemented using the versioning concept, under this approach

10. Since abrupt cut-over will terminate the old system abruptly and replace the old system with the new one the risk an organization needs to face is very high. There may be technical problems that are not identified in the testing stage because the system has not been fully operational in a real-life environment. In addition to the technical problems, system users may

not even be fully accustomed to the new systems. The new system may even face resistance because of that.

11. Since parallel conversion requires both the old and the new system to work simultaneously for a period of time, cost will become a major issue. Having two systems operating in the same time will increase the demand of the computer on which the systems are run. Thus, more resources may need to be allocated for that. Consequently, additional cost will then be attached to it.

12. - Alpha testing is also known as verification testing. Verification testing is done in a simulated environment with hypothetical simulated data. The test focuses on finding errors and omissions concerning the design and end-user specification.

    - Beta testing is also known as validation testing. Validation testing is done in a live environment using real data. Upon conducting validation testing, the testing will test the system for its performance, peak workload processing performance, and other testing such as human engineering testing, methods and procedure testing, and backup and recovery testing

13. System builders are the major player in installing databases. Application programmers will write special programs to transfer the data from the old databases to the new ones so that the new databases will be populated with real data.

14. The system analysts will need to make the end-user documentation or manuals of the system available for the users and start the training process. More importantly, system analysts must encourage the users to participate because they are the ones using the system. User involvement is vital. Thus, system analysts should facilitate the training process well so that users can really be comfortable using the new system.

15. Feedback is essential because through feedback we can identify inadequacies of the system which will need to be corrected and which will provide us with a benchmark for the new systems to be built in the future.

Problems and Exercises

1. Unfortunately, situations like this do occur at times, and too often the prevailing philosophy is “there is not enough time to do it right, but there is always time to fix it later.” As the system testing team leader, you have a responsibility to tactfully point out to the CEO that the cost of fixing problems

after implementation may very well be higher than the cost of delaying implementation. Further, the impact of a buggy system upon users may diminish or even destroy the ultimate success of the project; the good will of the users may be lost permanently. If you are unable to persuade the CEO to approve the time needed, your contingency plan is to adopt a risk-based testing strategy: identify the areas of highest risk in the new system and test those areas first.

2. This is a difficult situation that consultants and contractors may face at times. The organization for which you are doing the testing is paying your company for a certain level of quality and reliability. You have an ethical responsibility to ensure that the system you deliver meets that expected level of quality. If you can't meet that responsibility, you should ask yourself if that position is the right one for you.

3. There are several reasons that the programmers who built the system should not do the system testing. First, like proofreading, we tend to be blind to errors and mistakes in the work we've done ourselves. Second, there is a very different approach between stub and unit testing (which programmers do) and systems testing. In stub and unit testing, programmers (not necessarily the same ones who wrote the code) are testing to make sure the modules work. In systems testing, the testers are testing the system to see if they can break it. This is because if they don't, users will!

4. Systems testing teams should be composed of primarily systems analysts and business analysts, since this type of testing is not focused on the technical code, but on whether the application meets the business requirements. System tester skills include:
    - Attention to detail
    - Knowledge of testing techniques
    - Business knowledge of the system and the organization
    - Integrity and commitment to the organization

5. - False; they should be built first because the programs to be written will be dependent upon them as shared resources.
    - True: parallel conversion significantly reduces the risk of major or catastrophic damage if the new system does not work properly.
    - False; stub and unit testing are activities that can and should take place throughout the construction phase.
    - False; users should receive training in close proximity to the implementation date to ensure retention of knowledge.
    - True; systems development is a term that is sometimes used to refer to systems construction, but which is also used to describe the full development life cycle.

6. The intent of this exercise question is to give you the opportunity to write a portion of a user manual in order to appreciate the expertise required to write a user manual. Congratulations if your fellow students or co-workers found your writing to be understandable, clear and appropriately detailed; if not, remember that good business writers are always in demand, so it would be worth your time to develop your skills in this area.

7. The final authority on whether the system is operating correctly and ready for implementation is the system owner and users.

   The key input in to the implementation phase is the functional system from the construction phase.

   The systems construction phase is initiated (or triggered) when the physical design specifications are approved and the design phase is completed.

   Once the conversion to the new system is complete, ownership transfers from the project team to the end users.

   Migrating data from the old database and populating a new one is a complex activity that requires careful planning and execution.

8. No, users are essential to the implementation portion of this phase. In addition to being involved in system testing, they must also be involved in developing the conversion plan strategy, acceptance testing, training users, and in the actual conversion to the new system.

9. 1K, 2J, 3E, 4F, 5B, 6A, 7D, 8L, 9M, 10C, 11H, 12I, 13G

10. This question addresses a fundamental aspect of human interface design – to design a system that is so intuitive to use that, in theory at least, no user's manual is necessary. But is this objective achievable? There is no right or wrong answer to this question, only a well-reasoned argument one way or the other!

11. The PIER serves a number of purposes, particularly the following two:
    - Assuming that since the system worked immediately after implementation, it must still be working to expectations months later can be a serious mistake! The PIER formally reviews the system to determine if project objectives have been met, expectations are being satisfied, and no unanticipated problems have surfaced.
    - The PIER also serves as a "lessons learned" for future projects. It documents "what went right, what didn't go so well." PIERs can be an invaluable reference before embarking on a new project!

12. A poorly designed and constructed project will almost definitely fail, no matter how well-planned and well-executed the implementation effort is. In the end, the product must work; otherwise, to users, it is just hype!

    As for the opposite situation, a sound design and construction effort may carry a new system through a poor implementation effort. But a little planning can save a great deal of pain on everyone's part.

13. The project team and stakeholders should throw a party. The successful completion of a project is an event that does not occur every day, worthy of celebration.

Projects and Research

1. Students should find information readily available on the web for numerous products of this type, including those made by Rational and Mercury Interactive, who are leaders in this market. Students should be able to effectively compare and contrast their different features. Selections may vary, but expect most students to choose Rational because of its widespread use and familiarity.

2. Responses may vary, but the analysis should address the advantages and disadvantages in this situation of each of the different strategies described in the textbook. Also, given the scenario described, most students should recommend and provide a valid rationale for a parallel and/or location conversion strategy.

3. The purpose of this question is to provide the student with the opportunity to develop a simulated conversion plan and to receive real-world feedback from those with expertise in this area. The conversion plan should be consistent with the components described in the textbook.

4. As with the previous exercise, the purpose of this question is to provide the student with the opportunity to develop a simulated systems acceptance test plan and to receive real-world feedback from those with expertise in this area. The plan should use the general guidelines described in the textbook as a springboard, and should build on them with additional information and templates available on numerous websites.

5. The purpose of this question is two-fold: first, to have students gain experience in developing a training plan for end-users. Second, to research web-based training methodologies and to customize these methodologies so that they are applicable for end-user training.

6. The intent of this question is to expose the student to contingency planning for implementation, based upon both research and real-world experiences. Responses are open-ended, but should indicate that the student has thought through some of the possible things that might go wrong, and more important, has developed a contingency plan for dealing with the unexpected.

Minicases

1. Note to professor: Insist that students test each other's work with the intention of finding ALL of the flaws and making the prototype the best it can be. You will need to be clear that the class will not be graded on a curve – that if all prototypes are excellent, they can all get an "A." (Otherwise, students will 'forget' to test a portion of the other team's work!
2. Yes, this actually happened to a non-fictitious company. The lost revenue was not nearly as great, but there were many angry phone calls exchanged! Moral of the story: parallel conversion is the ONLY way to go on a website!
3. Make sure the students document (preferably in a table) ALL of the peer-recommended changes, and how *specifically* they addressed them in their revised prototype.
4. Best way to grade this is to hand the manual to a non-techie and show them the prototype. If the non-techie can figure out how to use the prototype without much trouble, then the manual passes a usability bar.

Team and Individual Exercises

There are no answers to this section.