CHAPTER THREE
GUIDELINES FOR DESIGNING AND IMPLEMENTING COMPUTER-BASED TRAINING FOR AVIATION MAINTENANCE

3.0 ABSTRACT

This report is an overview of computer-based training (CBT) hardware, software, and instructional design. It will assist aviation maintenance training and FAA inspection personnel in making informed decisions about CBT. A central part of this report is the description of a process for analyzing the instructional requirements for a specific task and then selecting a CBT program or designing and implementing a program from scratch. The intended audience for this report includes FAA managers, aviation training managers and instructors, and others interested in CBT design and implementation. After reading this report, the reader will have a better understanding of the managerial, instructional, and technological issues involved in selecting or implementing CBT programs.

3.1 INTRODUCTION

Broadly defined, a computer-based training (CBT) system is a combination of computers and special software for training and education. Within this broad definition, there are many different approaches, systems, and technologies. Their common goal is to transfer skills and knowledge from an expert to the student via a computer system in such a way that the knowledge will develop and/or improve performance on a set of tasks. This goal is the same as that of instructional systems in general, including instructional aids, text books, and instructors. What differentiates a CBT system from traditional teaching methods is that CBT can be interactive, dynamic, and individualized. CBT does not require one-on-one interaction with an instructor. The computer program can be designed to simulate a piece of equipment, to react to user actions, and to provide useful feedback when the user makes mistakes. Sections 3.2 and 3.3 of this report describe CBT in detail.

3.1.1 Purpose of This Report

This report is for

- FAA managers
- Aviation training managers
- Aviation training instructors
- Those who are interested in designing and implementing CBT systems.

This report

- Provides an overview of CBT hardware, software, and instructional design
- Helps aviation maintenance training and FAA inspection personnel to make informed CBT decisions

- Presents a process for designing and implementing CBT systems
- Describes the issues that instructional design teams should consider in designing and implementing CBT systems.

The reader does not need to be familiar with CBT design and implementation to understand and apply this report but should be familiar with aviation training issues. The author assumes that the reader's organization does not currently have CBT capability. Additional information relevant to the topics covered in this report can be found in Section 3.9, "References."

3.1.2 Overview of Part 147 Changes Related to CBT

Appendix A of FAR 147 describes curriculum requirements. Recent revisions to that section under "(c) Teaching material and equipment," state the following:

> The curriculum may be presented utilizing currently accepted educational materials and equipment, including, but not limited to: calculators, computers, and audio-visual equipment.

While this revision is loosely worded, it is interpreted to mean that CBT can be a reasonable part of the approved Part 147 school curriculum. Explicit guidelines for FAA inspectors have not yet been issued. However, it is generally expected that student hours spent with CBT will become an approved portion of school curriculum. This report will contribute toward making CBT an approved part of the school curriculum.

3.2 OVERVIEW OF COMPUTER-BASED TRAINING

Before an organization attempts to produce or use a CBT system, it needs to examine its need for training and understand the issues involved in CBT design, production, and use. This includes knowledge of what makes CBT successful or unsuccessful. This section describes the issues involved in deciding whether CBT should be applied to aviation maintenance training tasks.

3.2.1 When to Use CBT

There are no strict rules for deciding when CBT should and should not be applied. The ability to judge the appropriateness of any training technology comes from knowledge of the technology, experience with previous applications of the technology, and knowledge of the training task. Each training technology has its own peculiarities that make it appropriate for certain types of tasks and domains, and inappropriate for others. In the case of CBT , these peculiarities include the interactivity of training, the ability to include several information types, and the individualization of instruction. **Table 3.1** lists situations when CBT is generally appropriate.

On the other hand, CBT does not currently provide support for all types of training. The lack of realism in current computer interfaces and the difficulty of providing intelligent human-computer dialog means that some training is better accomplished with other methods. **Table 3.2** lists situations when CBT is usually inappropriate.

Table 3.1 When CBT is generally appropriate

- Textbook teaching methods are used in current course curriculum

- Task requires strong problem-solving skills
- Students need to understand dynamic behavior of equipment
- Real equipment is too expensive for training
- Task is dangerous
- There are not enough human instructors available
- Flexibility of training schedule is desirable
- Training needs to be delivered to multiple, decentralized sites
- Students need extensive practice and review to accomplish task
- Students have wide range of experience and backgrounds
- Consistency of training is important

Table 3.2 When CBT is generally inappropriate

- Significant psychomotor or sensory skills are required for task
- Training requires significant instructor/student interaction
- Aesthetic judgement is required for task evaluation
- Extensive team skills need development
- Few students require training
- Training needs to be implemented quickly

As shown in Figure 3.1, there are two reasons for applying CBT to a particular task: reduced cost and improved performance (Kearsley, 1982). In the case of reduced cost, the goal is to provide training more cost effectively than the current method while maintaining the quality of training. Cost savings can come from reduced work force requirements, decreased travel cost, lower training equipment costs, and a decrease in the time required to attain knowledge or skill proficiency. For example, a CBT program that teaches a new maintenance procedure could be more cost-effective than teaching the procedure with an instructor because of money saved in travel and per diems. Improved performance, the second reason for applying CBT, results from the high-level of interactivity possible with graphical computer simulations. For example, a CBT program could improve technicians' troubleshooting abilities by allowing them to practice and improve their diagnostic skills. A CBT program can provide realistic simulation that rivals that of on-the-job experience, without the danger, equipment, and time required for on-the-job training.

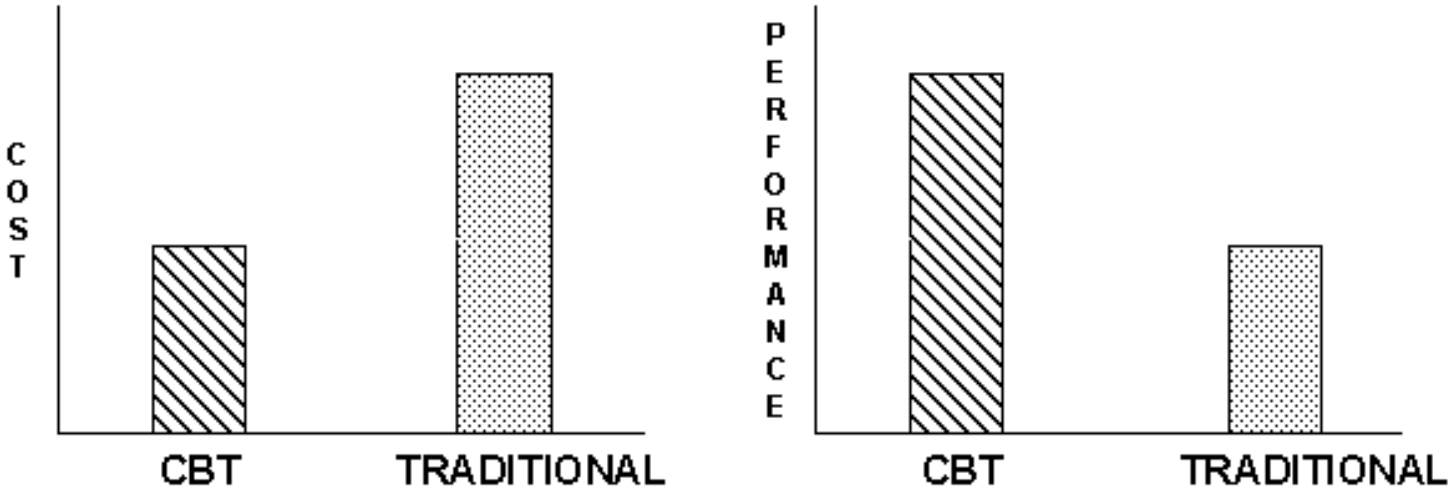

Figure 3.1 CBT goal to lower cost and increase performance

There are financial tradeoffs to consider when comparing CBT to traditional training. Figure 3.2 shows a generic trend comparing the development and delivery costs between the two methods. The initial development cost for CBT is often higher than for traditional instruction. However, over time the CBT delivery cost is likely to be less than that of traditional instruction. Financial considerations are also discussed in Section 3.4.1.3.

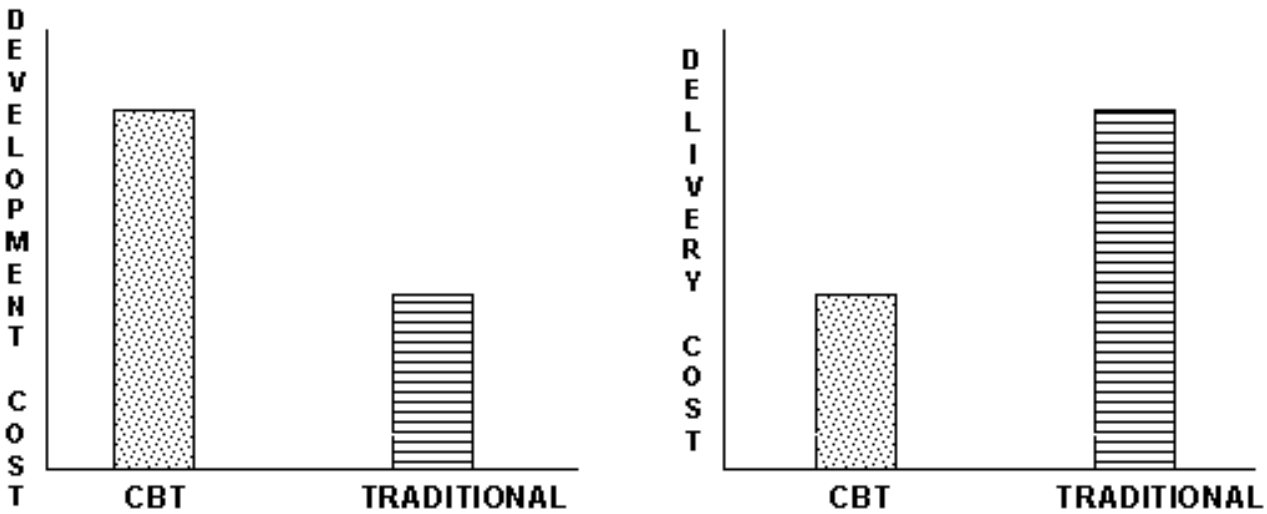

Figure 3.2 Comparison of development and delivery costs for CBT and traditional training

3.3 OPTIONS

There are many decisions to make in designing and implementing a CBT system. The selection of approaches and technologies should be based on the organization's instructional needs and budget. This section describes factors that must be considered when creating CBT programs.

3.3.1 Instructional Approach

Depending on the type of information and knowledge being taught to the student, there are usually several appropriate instructional approaches. For example, to teach the rules of the road, a standard present-and- test approach is appropriate. Actual driving (or a simulation) is appropriate for teaching the physical and coordination skills necessary for safe driving. This report lists several common approaches to instruction and gives examples of when each is appropriate. Note that a CBT program may combine several of these elements. For example, a "simulation-based" CBT could include an "intelligent training" component. Table 3.3 lists four instructional approaches.

Table 3.3 Instructional approaches

	What is it?	When to use it?
Linear	Resembles book reading	Teaching facts and procedures
Simulation-Based	Simulates equipment or tasks	Teaching how to perform tasks
Intelligent Tutoring	Provides human-like response to user actions	Teaching a complicated task that requires reasoning
Microworld	Simulates context of problem solving	Teaching relationships between actions and outcomes

Table 3.3 Instructional Approaches

3.3.1.1 Linear Training

The linear training method of CBT presents the material in much the same way as a book. Users can "step" forward and backward through the material, and possibly jump to other topics and subjects. Typically, the instructional design is taken from a book or lesson plan that has been used to teach the course in the past. Linear training differs from a book in that the program can use multiple types of presentation methods, including graphics, audio, and video. Multiple types of media (multimedia) can make such a CBT system more interesting and instructive than a book or lecture.

Linear training is most appropriate when the material being taught consists of procedures, lists of items, or regulations that the student must memorize. One example of linear training is a CBT system that teaches the procedures for starting up an aircraft's engines. This type of task requires that the technician follow procedures in a specific order, so linear presentation of the material is appropriate. The testing unit of the CBT could then ask the user to order the procedures in the correct sequence. Linear training is usually the easiest to use, and it is also the least complex to design and implement. CBT that uses linear training has to be carefully designed to hold the user's attention, since linear presentations can quickly become monotonous.

3.3.1.2 Simulation-based Training

A simulation-based CBT system differs from a linear training system in that the former attempts to simulate some type of task through dynamic graphics. The software provides a realistic imitation of the necessary equipment and activities and behaves like the "real" world. For example, the CBT may require the student to troubleshoot a piece of equipment by inspecting, testing, and replacing its components. By simulating only the important features of a task, simulation-based CBT can provide cheaper, faster, and safer training than a real system, while requiring limited supervision from an instructor.

Simulation-based training should be used when the training task involves the manipulation of complicated equipment or when the equipment itself is too expensive to use for training. Examples of such training systems are a CBT that trains for avionics troubleshooting skills or a CBT that teaches the operation of NDI equipment. The way that a student uses a simulation-based system is much less constrained than the way a linear CBT works, since the student can perform the actions in any order.

3.3.1.3 Intelligent Tutoring

An intelligent tutoring system (ITS) mimics the instructional strategies of an instructor or domain expert. An ITS can give advice, provide feedback, and explain mistakes. The levels of intelligence in ITSs vary from "canned" text feedback to expert systems that understand the domain being taught. By automating some of the assistance that instructors usually have to repeat several times, ITS can provide consistent training to a large number of students.

An ITS cannot replace the role of the instructor because computer programs currently lack the intelligence and flexibility of human trainers. Computer intelligence in training should be seen as an instructional aid. An ITS can provide help for the simple, commonly asked questions, while the instructor provides the answers for the harder questions. These systems can perform some of the simple training and help tasks of the instructor, but the tasks requiring large amounts of interactivity and intelligence should be left to the instructor. By automating these tasks, the instructor has more time to answer the difficult questions and provide individualized instruction to students who need more help.

3.3.1.4 Microworld Training

Instead of simulating a piece of equipment or a particular task as a simulation-based trainer does, a microworld trainer models an entire job for a particular worker. The microworld trainer attempts to give a realistic context for problem solving, thus increasing the chance that the user will remember what he or she learned in the training while on the job. An aviation technician microworld CBT, for example, would simulate the cognitive tasks of troubleshooting, and allow the technician to replace parts and test the equipment.

Microworld training is most appropriate when the goal is to teach cognitive skills. Most text-based training systems attempt to teach the user a set of facts; a microworld trainer imbeds these facts in real- world problem solving. Microworld training provides the user with a context for problem solving, making it easier to remember when a similar situation occurs on the job.

Another advantage of microworld training is that it can be extremely engaging when designed correctly, thus motivating the student to use the CBT. Microworld computer games (such as SimCity and) have high entertainment value, in comparison with other computer games, because the player has full control over the course of the game.

Remember: The instructional approaches described above are not mutually exclusive. Combine them to meet the students' instructional needs.

3.3.2 Presentation Formats

The training and instructional analysis provides a functional description of what information the CBT must provide to the user. Presentation media affects a CBT's cost so the media should be selected based on instructional criteria, rather than any aesthetic judgements or preferences. Common presentation formats are summarized in Table 3.4.

Table 3.4 Presentation Formats

	When to Use	Guidelines for Use
Text	Identify and describe processes, objects, and procedures	• Limit word use, be clear • Use large fonts and readable colors
Graphics	Show what equipment looks like or how a system is organized	• Make as simple as possible and do not show unnecessary objects • Consider display resolution of computers
Animation	Explain a process or demonstrate the steps of a procedure	• Makes the program more engaging • Do not make longer than necessary • Give user control over playback
Audio	Add realism, increase entertainment factor, or communicate long text passages	• Do not overuse; have a reason for using it • Allow user to control volume, turn off
Video	Describe procedures or increase entertainment factor	• Give user control over playback • Match purpose with video quality

Table 3.4 Presentation Formats

3.3.2.1 Text

Text is the most common CBT presentation format, since all computers support text. Text can be used to identify and describe processes, objects, and procedures. While audio narration has a fixed rate of playback, text allows the reader to absorb information at his or her own pace. One drawback of text is that the low resolution of most computer monitors makes it difficult to read long text passages. CBT designers should take this into account by shortening the length of text and deleting unnecessary words and sentences.

3.3.2.2 Graphics

When a CBT program needs to show what a piece of equipment looks like, or how a system is organized, a graphic is the best presentation method. Graphics can be pictures or line drawings of equipment or schematics showing connectivity and functionality of components. The CBT team must consider the resolution of graphics required for effective instruction, as this partially determines the type of computer display necessary to support the program. The required resolution is a function of the purpose of the graphics: teaching for recognition requires high resolution, while simple schematic drawings can be displayed in lower resolution. In cases where aesthetics or entertainment value is important, higher resolution may be desirable.

3.3.2.3 Animation

Motion is an effective method of drawing attention to a particular item. The simplest way of displaying motion is through animation, which is the movement of simple graphical objects along simple paths. Animations differ from video in that they capture the general content of an event while discarding some information, thus making it easier for the user to understand the relevant issues. An animation can be used to explain a process or to demonstrate the steps of a procedure. Examples include animations of flows in electrical and hydraulic systems and animations of the installation procedure for an avionics component.

3.3.2.4 Audio

Audio in CBT can add another level of realism to the training system, give more entertaining feedback, or make the communication of long text passages much simpler. CBT audio can be divided into three classes: narration, equipment sounds, and musical accompaniment. Narration allows the user to listen to a description while watching an animation or video; this is not possible with text annotations. Machine sounds such as jet engines are usually not needed for instruction but may increase the entertainment value of the program. Musical accompaniment makes the program more interesting or adds emphasis to a point. Before adding audio to a CBT program, remember that an audio sound board is required to play audio on PCs, and audio files can greatly increase the program's total size.

3.3.2.5 Video

Like animation, video can be used to describe a process or to show a procedure. Video differs from animation in that it is a more accurate representation of the "real world" and usually has an accompanying soundtrack. Since video is more realistic than animation, it is usually better for describing procedures such as test or installation steps that a technician will perform on the job. The viewer more easily understands the context of actions shown in a video than in an animation. There are several methods for displaying video on computers, with varying degrees of resolution and cost (Pearce, 1993). All current desktop full-motion video technologies require some type of high-end hardware and large amounts of data storage.

Remember: It is important to decide early in the design process which data types the CBT will use, since this partially determines the training application's hardware and software requirements.

3.3.3 Integrated Training and Job Aiding

Another important issue to consider before designing a CBT system is whether technicians could also benefit from a computer-based job aid. A job aid differs from a training system in that a job aid helps the technician when the task is performed, not before it is performed. This approach to training can also be called "just in time," borrowing from current "just in time" inventory and production management techniques. Some training is still necessary with a job aiding system, since workers need some background knowledge before starting to perform a task.

Because the knowledge and software required for training systems and job aids are usually very similar, it is more cost-effective to design a system that serves dual purposes than to design two separate systems independently. Integrated training and job aiding also implies active learning and on-the-job training. The chance of the user remembering an important fact or lesson is greater when the user is actually performing the job than when he or she is sitting in the training room. A job aid also decreases the need for "up- front" training, since the worker learns how to perform a task immediately before doing it.

3.4 HOW TO DESIGN AND IMPLEMENT COMPUTER-BASED TRAINING

The following section describes a process for designing and implementing CBT. The authors have used this process successfully. It maximizes the likelihood of producing successful CBT systems. Figure 3.3is a flow chart of the steps in and outputs of this CBT production approach.

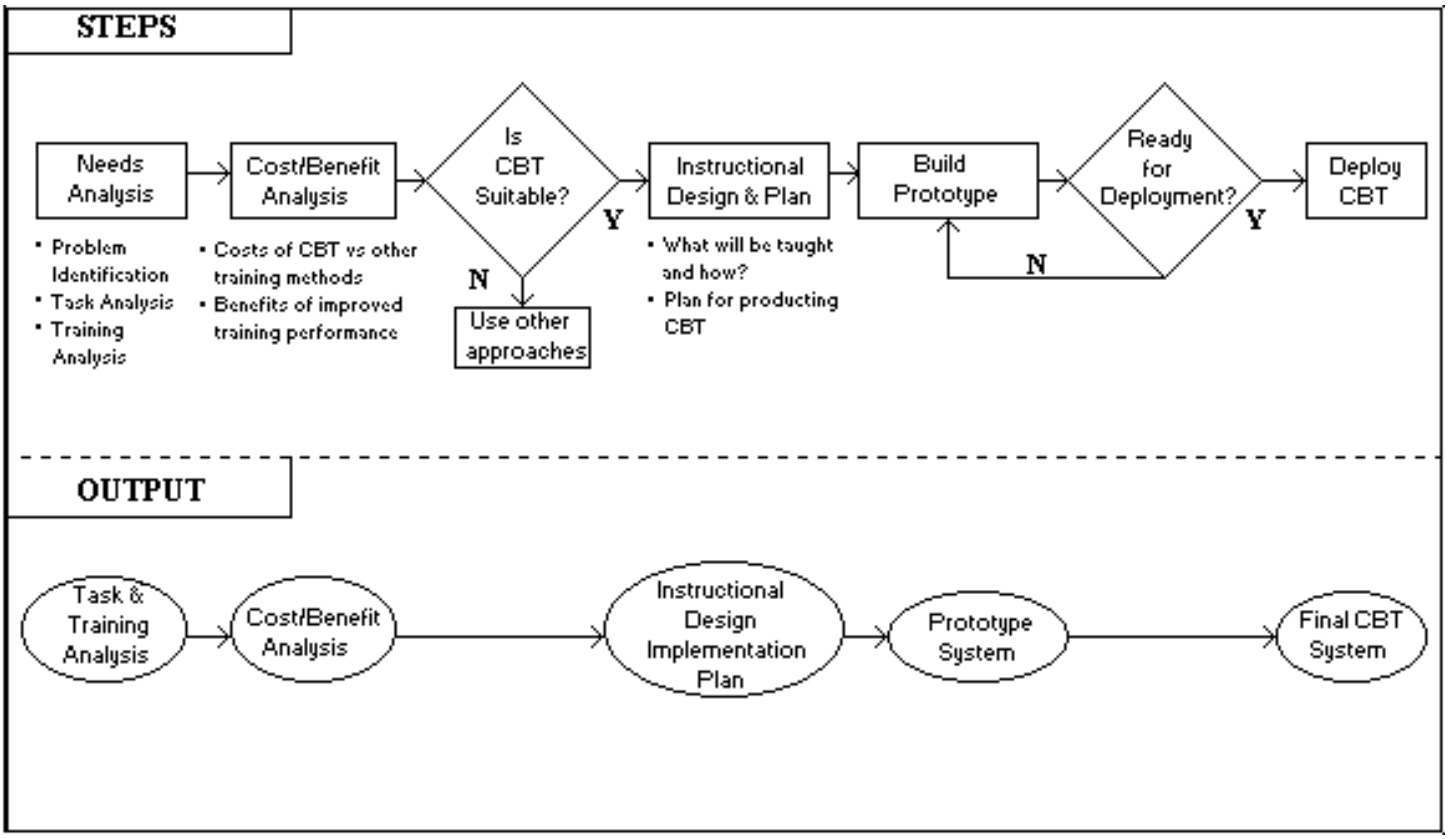

Figure 3.3 Typical CBT development process with outputs

3.4.1 Needs Analysis

The first step in designing a CBT system is to decide whether the system is really needed for a specific task, and to describe the expected benefits of implementing a CBT system. Note that this step is not specific to CBT; a needs analysis is necessary for all training systems and job aids. If a need is incorrectly identified or the nature of the problem is not understood, then the final product probably will not be useful. The needs analysis consists of three main steps: problem identification, current training methods analysis, and cost/benefit analysis. Each step is described below.

3.4.1.1 Problem Identification

The first step in needs analysis is to identify a task that needs improved performance or lower training costs. This can be done informally through interviews with technicians and managers. A more formal evaluation would include a description of the technicians' tasks, how long it takes technicians to perform these tasks, and any "bottlenecks" that delay completion. The final product of this step should be a list of technicians' tasks that may benefit from improved training methods.

Remember: The two reasons for using CBT are to increase efficiency for a task and to increase effectiveness of training.

3.4.1.2 Current Methods of Teaching the Task

The second step in needs analysis is to perform a training analysis to determine how the training for the task is currently conducted. This step helps determine if an improvement in the current method is possible and to identify strengths and weaknesses of present training methods. Once the current process is described explicitly, it is much easier to critique it and find its weak and strong points. Such a description should include what is taught, who teaches it, how long it takes, how it is taught, what resources are needed, and the students' performance before and after training.

The training analysis identifies one of the following possibilities: no formal training exists, training exists but could be improved, or current training is sufficient. Typically, workers learn many of their tasks on the job. When there is an existing training program, there are two more possibilities: the training could be made more effective (thus increasing technicians' performance and productivity), or efficiency could be increased (which decreases the training cost).

3.4.1.3 Pre-Production Cost/Benefit Analysis

The final step in needs analysis is to determine if training for the specified task will give an adequate return on investment. This is one of the hardest tasks in CBT production, because it is difficult to:

- Measure benefits of intangibles, such as safety and customer satisfaction
- Predict the actual performance improvement for a system that has not yet been built
- Predict the cost of building CBT systems (and computer programs in general).

Given all these unknowns, it is still possible to generate a rough estimate of a training system's effectiveness before it is produced. It is important to remember that these are rough estimates, so take this into consideration when planning resources and deadlines.

In deciding if CBT is appropriate for a specific task, look at other training methods and compare their costs and benefits. There are several methods for doing this, but this report will only mention them in passing, as most of the methods involve complex economic and mathematical analysis. One method is to compare the total "life cycle" costs of the CBT to that of other systems (Kearsley, 1982). Typically the costs are broken down into three categories:

- Research & development/start up costs - associated with planning and designing CBT
- Transition costs - one-time costs of implementing CBT hardware and software
- Steady state costs - continuing costs (such as maintenance) of using CBT.

The result of this step should be a description of the CBT system's goals and an estimate of its benefits (either improved efficiency and/or decreased cost). The goals provide a high-level summary of what the CBT system will do for the users.

3.4.2 Job Task Analysis

Job analysis is a process used to break a job into tasks and rate the tasks for further analysis. Task analysis is a process to determine how a task is performed and the factors that affect performance. Consequently, job task analysis (JTA) is a method used to identify what tasks are included in a particular job and how these tasks are supposed to be done. JTA identifies the knowledge, skill, and activity requirements necessary for job performance. The goal of the JTA is to explicitly describe the job so that the workers' efficiency, effectiveness, and error rates can be improved. There are many different methods for performing JTA, ranging from detailed formats for describing each step in performing a task to more informal observational methods (Woods, 1993).

Formal JTA methods are used during system design or when there appears to be serious problems with the design of a system or the procedures used to operate it. In most cases, JTA need not be formal. JTA for instructional design can be as simple as a one-page description of what the workers do to perform the task. Particular attention should be paid to errors that repeatedly happen on the job and to inefficiencies that could possibly be eliminated through improved training.

Remember: Each organization will have to determine its acceptable level of JTA. JTA is merely a "means" to the "end" of developing CBT. The JTA process must not become the product.

3.4.3 Instructional Design and System Planning

Once the JTA is finished, the instructional design of the system must be created. Instructional design is concerned with understanding, improving, and applying instructional methods to specific problems (Reigeluth, 1983); it describes *how* certain material will be taught. This differs from curriculum design, which describes *what* will be taught. The goal of instructional design is to choose the best methods to achieve specific improvement in knowledge and skills for a given group of students. The description should summarize the training process to be implemented in the CBT program, including

- What the CBT will teach (its instructional goals)
- Knowledge students are assumed to know before using the system
- The instructional approach
- Structure of lessons, exercises, tests, etc.
- Types and number of questions
- How the information will be presented (text, graphics, video, etc.)
- How the CBT will be used with other instructional methods (instructor, books, etc.)
- How to evaluate the students' knowledge.

There are several common approaches to instructional design. Which approach is appropriate for a particular project depends on the training task. See Section 3.3.1 for more information on the various approaches to instructional design.

Once a particular instructional approach is chosen, the CBT team has a better idea of the time, skills, and resources necessary to complete the project. For example, simple training systems with text instruction do not require much effort to program and test; a large training project with complex equipment simulations and a multimedia presentation demands much more effort. The CBT team manager should apply standard planning methods, including

- Developing a program time line and Gantt chart
- Planning resource allocation, including people, computers, software, training, etc.
- Setting milestones for the development cycle.

Remember: The instructional design is one of the most important components of a successful CBT system. Poor instructional design will lead to poor training and possibly poor job performance.

3.4.4 Prototype System

Once the instructional design is finished, system implementation can begin. This is done by producing one or more prototypes of the CBT program. A prototype is a tool for critiquing the design of the system. The demo system can be evaluated by the designers, content experts, managers, users, and programmers to ensure that they all agree on the system's design before full-scale implementation begins. If there is not agreement that the prototype design is good, then it is unlikely that the organization will accept the final system.

The prototype demonstration is best done on a computer since this provides the interactivity that other methods lack. The demonstration program should include the "main" features described in the instructional design, with the secondary features (for example, student logging and multimedia) omitted. Also, some of the prototype computer program code can be used in the production version of the CBT program.

Although computer-based prototyping methods require more up-front effort than other approaches, they are often more efficient in the end. One advantage of computer-based prototyping is the ease with which the users can participate in program design. A user can suggest changes to the program, and the changes can be made while the user watches. Another advantage is that the effort which goes into building the prototype can be applied to the actual CBT programming by using the prototype as the basis for the CBT program. The use of authoring systems in building CBT (covered in Section 3.6.3.2) makes the transition from prototyping to programming more manageable.

There are several prototyping approaches that do not require computers, and some find these methods easier to use. One method of producing a quick demonstration without computers is to produce a paper "storyboard" of how the screens will look. Each "frame" of the storyboard includes a rough sketch of what the screen will look like with a caption that describes the screen's purpose. Storyboarding can also be done on an erasable board, so that many CBT team members can participate in the design. A storyboard created before the computer prototype can act as a guide for the programmer.

3.4.5 Rapid Prototyping Software Development

The rapid prototyping approach to software production grew out of the observation that many computer programs do not meet the needs of the intended users. This often is the result of a divergence between what users need and what programmers think the users need. Often, users do not know what they want or do not want until they see it. Rapid prototyping helps solve this problem by including users in the software design process and by using their feedback to drive changes and improvements to the program (Sewell & Johnson, 1990).

The last two steps described in this CBT production process (prototype system creation and evaluation) are used as the basis for rapid prototyping. The instructional design is a definition of what requirements the designers want the software to meet and a high level description of how the program will meet the requirements. From this analysis, the programmers create a prototype program that can be critiqued by instructors, students, and administrators. The programmers then make changes to the software based on the users' feedback during the evaluation. The evaluation should include checks for

- Correctness of the information
- Ease of use of the program
- Spelling and grammar mistakes
- Graphical layout, user of color, and other aesthetic issues
- "Bugs" (programming errors) in the CBT program.

To ensure that the reported problems are addressed, it is best to have a formal reporting scheme. Such a scheme can be as simple as a written log of each problem found, the person who found it, the solution to the problem, and the person(s) who fixed it. Once all of the bugs have been fixed in the current version of the CBT, it should be tested again to ensure that previous problems were solved and that no new problems were introduced. This iterative process continues until the organization is satisfied that the software is suitable for its intended purpose. Because this step in CBT production is so unpredictable, set aside enough time to address both large and small problems that come up during the evaluations.

3.4.6 Evaluation of CBT System

Evaluate the CBT system to determine if any changes are needed to fulfill the goals of the CBT system and to provide data for future CBT systems. In the first case, the evaluation examines the instructional features of the CBT system and how the students use the system. Usually, any problems found may not have been known at the time the program was written and may arise from design decisions. For example, students without much computer experience may not understand how to use a control on the CBT program's interface that is familiar to experienced computer users. The other purpose of an evaluation, to help in developing future CBT systems, has the goal of using what was learned during the design and implementation of one CBT system to assist in the creation of other CBT systems. The data for this evaluation may include formal experiments or informal surveys of users' experiences with the training system.

One drawback of evaluations is that they can be very expensive. While one or two people can build a CBT system in several months, it takes numerous experimental CBT users to provide statistically significant data. Depending on what type of data are being collected, evaluation can require anywhere from several hours to several weeks of a CBT user's time. Also, it is difficult to collect "on-the-job" data, so written tests that may not measure actual learning and job performance improvements are often used as substitutes. Because of the expense of some types of evaluations, the CBT team should formulate specific issues that they want the evaluation to examine. When producing an evaluation, the CBT team should

- Develop a hypothesis about the training system(s)
- Collect data relevant to the hypothesis
- Analyze the data with respect to the hypothesis
- Apply the outcome of the evaluation to current and future training systems.

There are several common models for evaluating training. Models vary in the types of data needed for the analysis and the types of information that they produce. The appropriate method also depends on whether the CBT system was built to save money, to improve performance, or both. Sometimes a combination of models is appropriate to generate the information needed to make future decisions. Table 3.5 summarizes the types of evaluations commonly used on CBT systems.

Table 3.5 Evaluation Methods

	Purpose	When to Use
Questionnaires and interviews	Collect information about users' opinions	After CBT is finished
Content verification	Ensure that information is correct	During development
Cost/benefit analysis	Determine the CBT's overall cost effectiveness	After CBT is finished
Usability testing	Evaluate the quality of the interface	During development

Table 3.5 Evaluation Methods

3.4.6.1 Questionnaires and Interviews

Cost and learning evaluations only tell part of the story. To judge users' opinions concerning a CBT system, the most appropriate tools are questionnaires and interviews. A questionnaire attempts to determine what users like and do not like about a CBT system. Make the questionnaire short (two pages or less) and simple, so that it can be filled out in a few minutes. The data can then be analyzed statistically to determine the system's strengths and weaknesses.

Interviews also attempt to collect data about users' opinions of a CBT system, but they are less formal than written questionnaires. An interview usually consists of a set of questions for the users and instructors, and they can also allow the subject to discuss unanticipated issues. Although interviews require preparation, data evaluation is often more difficult and subjective because the answers and comments tend to be more open-ended. Consequently, interviews are better for identifying specific problems with a CBT system than for determining general trends in acceptance and satisfaction.

3.4.6.2 Content Verification

The most important component of a CBT system is its content, since it is content knowledge that is being taught to the users. If the content is incorrect, then the students will not learn the correct information. This may lead to poor and/or unsafe job performance. A content validation verifies that the information contained in the CBT is correct.

Content evaluation is commonly accomplished by showing the CBT to experts and instructors and having them check for omissions and errors. Ideally, the people doing the content evaluation should not be the subject matter experts who provided the knowledge for the CBT program. It is possible that the subject matter experts have misconceptions about the domain. If multiple experts are to evaluate the content, it is likely that there will be conflicts and contradictions in the advice and suggestions they offer. Therefore, when analyzing the results of the content evaluation, differentiate between different styles of problem solving and training and significant errors in the domain knowledge.

3.4.6.3 Cost/Benefit Analysis

Cost/benefit analysis combines two separate measurements to determine the return on investment for the CBT. The cost analysis measures the total cost of designing, producing, and distributing the CBT. This data can be gathered from records of time and money spent on development. The benefit portion of the analysis focuses on changes in worker's performance. Benefits are measured from changes, or estimates of changes in on-the job performance.

If the benefits analysis concludes that the training is not effective, then other types of evaluations are necessary to pinpoint the source of the problem. This is because a benefits analysis "lumps" the good and bad features of a CBT into one measurement. Some common measures of success of CBT systems (and training systems in general) are:

- Higher achievement and productivity level
- Improved quality of work
- Decreased student training completion time
- Increased motivation level
- Increased safety and a lower accident rate.

The more abstract the measure of training success, the harder it is to correlate it with actual CBT features. For example, if there is an increase in productivity after the introduction of a CBT system, the increase may or may not be a result of the CBT. It is usually easier to eliminate other explanations such as the use of new equipment or new accounting methods than to prove that changes in performance are a result of the introduction of CBT.

3.4.6.4 Usability Testing

A usability test is conducted to ensure that students can operate and understand the system. This is necessary because the usability of a CBT system is an important factor in determining whether or not the users learn. Even if the information in the program is correct, users will not focus their attention on the lessons if the interface is poorly designed. For example, novice computer users may not understand conventions known to more advanced computer users. CBT designers need to keep this in mind when designing the interface. It is advisable to make the interface as simple as possible and to avoid cluttering it with unnecessary text and graphics.

Frequently, usability testing is done by observing a few potential users while they operate the program. It is best to have users who have not used the program before and who do not have much computer experience. These users will report problems such as not understanding what to do next. All users should be encouraged to report any difficulties before they start the usability testing. To identify other types of problems, it is best to watch the users and see what types of mistakes they make, keeping interruptions to a minimum. The problems in the interface should then be listed so that the CBT designers can discuss them. Once the interface problems have been fixed, the CBT team should conduct another usability test to ensure that the problems have been adequately fixed and that new problems have not been introduced.

Remember: Evaluation is part of the rapid prototyping software development method. The evaluations should be done frequently during CBT design and development. Ongoing evaluation eliminates surprises at the end.

3.4.7 Maintenance and Updates

A CBT system is not finished after the programming and evaluations have been done. The system needs to be kept up-to-date with changing technology, equipment, and procedures. For example, the maintenance procedures may change for a particular piece of equipment. If an existing training system refers to the old maintenance procedure, it will have to be updated to the current method. Also, students using a CBT program in the classroom usually uncover bugs that were not found during program testing. Instructors commonly request that new features such as automatic grade reporting be added to CBTs after they begin using them. These costs are called "steady state" costs because they remain somewhat constant after the CBT program has been finished and is being used.

Other "incidental" costs are rarely planned for, but commonly arise. Computer hardware maintenance costs vary widely, depending on the computers' quality and the amount of use. Public-access computers have higher levels of wear than computers used by individuals; so the budget should account for cleaning, maintenance, and repair. If the lab is to remain open for extended hours, security and lab personnel have to be budgeted. Also, such costs as those for printing paper and writing documentation are necessary for some types of CBT.

3.5 ASSEMBLING A CBT TEAM

One of the most important considerations in producing a successful CBT system is choosing the team members. Designing a CBT system requires skills in diverse areas such as training, graphics, computer software and hardware, and usability. The CBT team needs all the skills above, since the lack of even one skill can lead to significant problems with the CBT program. If the team does not have these skills, it should seek training in the unknown areas or hire consultants to fill the gaps. Sometimes one person may possess two or more of the skills, so the team does not always have to be large. Figure 3.4 shows the makeup of a CBT team.

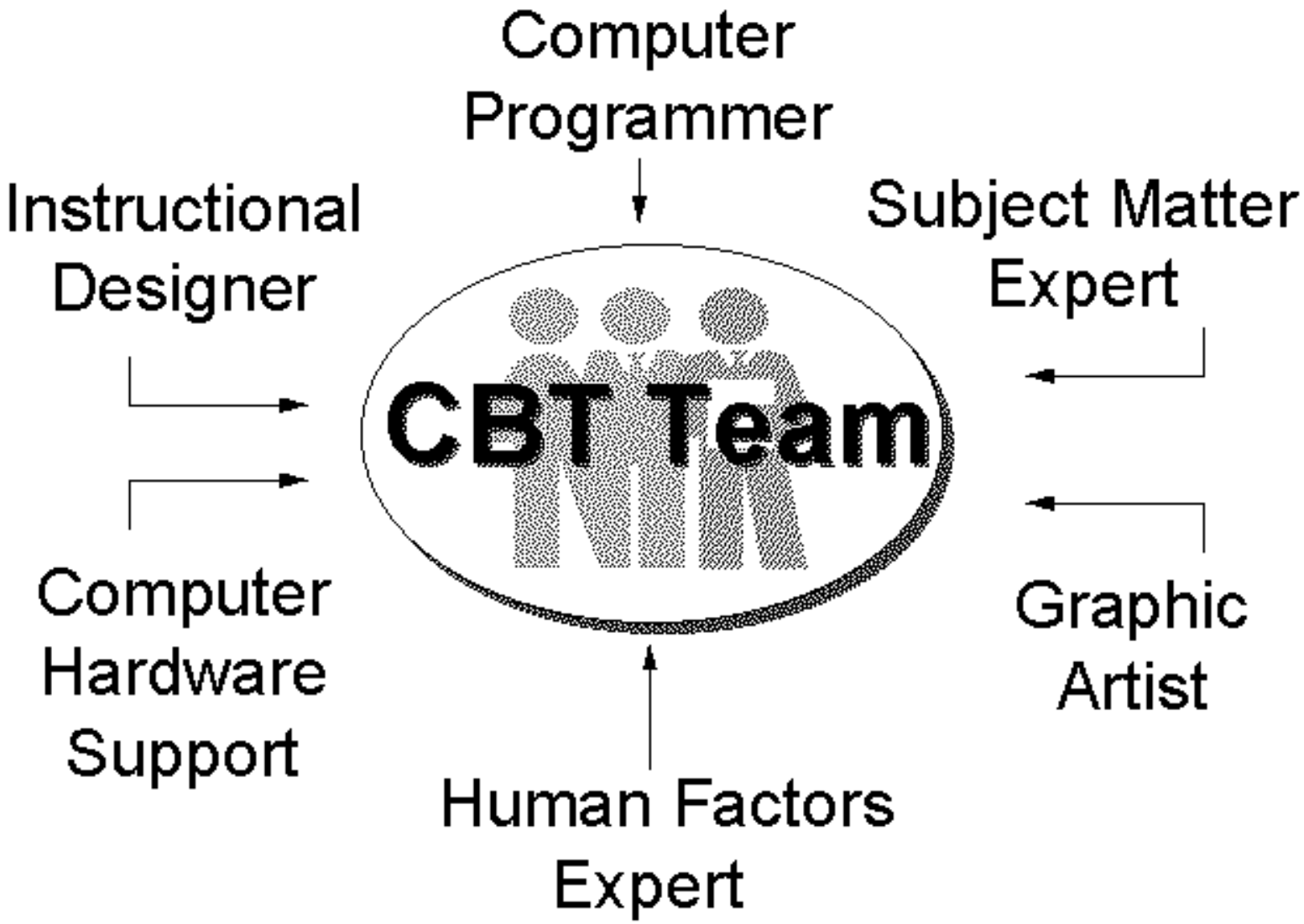

Figure 3.4 The CBT Team

3.5.1 Instructional Designer

The instructional designer applies knowledge of training, learning, psychology, and other training systems to the design of the CBT system. Most instructional designers have advanced degrees in education or psychology and have the ability to apply their knowledge to the design of training systems. This person should also understand the training domain, the jobs and tasks of the workers, and how training principles and past experience apply to the current training system. Because instructional design is the most important consideration in CBT design, the instructional designer is often the team leader.

3.5.2 Subject Matter Expert

The subject matter expect (SME) is key to the success of a CBT project because he or she is the source of the knowledge and information that the CBT system will teach to the students. If the SME does not understand the domain well enough to explain it to the instructional designer, then the chances of building a successful CBT system are slim. Therefore, the SME is often an instructor or someone with much on- the-job experience. Johnson (1988) says that instructors are the best SMEs because they understand the subject and instructional principles and also know how to best explain the domain to programmers and other CBT developers.

A common problem in building CBT systems is difficulty in finding and retaining a knowledgeable SME. In aviation, as in most other domains, experts are scarce and are often committed to other projects. Experts may be pulled from a CBT project for more pressing tasks, leaving the CBT team with the difficult task of trying to bring a new SME up to speed on the project. In this case, there is a familiarization period before the new SME can contribute to the project, thus delaying any new work on the CBT. This problem can be avoided by having two or more experts working on the project, even if one or more of the additional experts are only secondary contributors. This approach has the advantage of allowing more than on person to verify the domain representation.

3.5.3 Computer Programmer

The programmer implements the instructional designer's training plan. An important part of a programmer's job is the ability to design a program that meets the constraints of time, money, performance, and computer hardware. The programmer should have some understanding of instructional design and software usability. The programmer must also be able to take users' feedback and convert it into changes making the software easier to use and more understandable.

Programming usually takes the largest portion of time in implementing a CBT system, and large projects require more than one programmer. Programmers' work load can be reduced by ensuring that they have adequate software tools, computer hardware, and design feedback from the rest of the team. Experience is the most important factor in determining a programmer's productivity; an experienced programmer can produce considerably more software than an equally intelligent but inexperienced programmer.

3.5.4 Computer Hardware Support

The computer hardware support person ensures all the computer hardware and software is working properly. If something is not working properly, the hardware support person should know how to test it and how to resolve hardware malfunctions. Another large part of the job is maintaining software configurations, making sure that the computers have the latest version of the CBT software. This person should be able to help the students use the software and to answer basic questions about the training program.

3.5.5 Human Factors Expert

When students cannot easily use a CBT system, chances are that they will not learn much. When students' attention is drawn away from content, they concentrate on trying to operate the software. The human factors expert ensures that the system is understandable and easy to use. The human factors expert can also provide design guidelines pertaining to program behavior, screen layout, colors, and fonts. It is important that the human factors expert be involved early in the design process so that he or she can critique the overall CBT approach. A human factors expert should have experience in the psychological issues of interface design and have practical experience in computer program design.

3.5.6 Graphic Artist

A good graphic artist is often vital to the timely completion of a CBT project. Graphics production is time-consuming. If the graphics are on paper, they must be scanned into the computer and then cleaned up with a paint program. If the graphics need to be drawn from scratch, the instructional designer or subject matter expert should sketch them so that the graphic artist does not have to guess at what is needed. The graphic artist should be familiar with issues of color selection, screen layout, design for readability, and usability, and should have some previous experience in designing graphics for the computer. When designing the screen layout and graphics, it is important to keep in mind the graphic limitations of the target computer configuration. For example, when graphics designed to be displayed on systems with 256 or more colors are displayed on computers that can only show 16 colors, the resulting graphics are not understandable or attractive.

Remember: The CBT team does not need one individual for each of the described tasks, but all of the skills should be included on the team.

3.6 HARDWARE, SOFTWARE, AND SUPPORT

This section covers a variety of topics related to the budgetary and commercial issues in CBT. The material is not technical and presents enough information to allow managers to make informed decisions.

3.6.1 Hardware

Since they first became widely available, there has been a constant increase in the power of personal computers (PC). This increase in power has been followed by a constant increase in the computing power requirements of PC software. There is now much personal computer software that will not work (or works too slowly) on computers that are more than a few years old. Because of the high cost of installing and maintaining new computers, many older (and less powerful) computers are still in use.

When designing a CBT system, the CBT team must consider the type of computer that will be used for running the CBT program. If the hardware does not support the CBT software, then the students will not be able to make full use of the training. Computer hardware can vary along several dimensions, so the CBT team must consider all these dimensions when they design the CBT system. This section describes the various dimensions and also explains when special hardware may be needed, and what the hardware will add to the instruction.

3.6.1.1 Processor Type and Speed

A PC's processor determines how "powerful" the computer is and how quickly it can do its computations. Because of improvements in processor technology, modern desktop computers are roughly 100 times more powerful than PCs of ten years ago. Processor speed is analogous to the thrust of an aircraft's engines, which determines the power of the aircraft. Some text-based applications can be used on an Intel 8088 computer, the first generation PC processor, although few of these are currently manufactured. The next step up is the Intel 80286 and 80386 processors which usually have clock speeds between 15MHz and 33MHz. For the most computer-intensive programs, the Intel 80486 with a 33MHz or higher clock speed is required.

3.6.1.2 Computer Memory

A PC's memory determines how much data it can work on at a single time. For text-based computer programs, 640 kilobytes (kb, or about 1,000 "words" of computer memory) is an acceptable amount of memory. For more complex programs, between 2 and 8 megabytes (Mb, or about 1,000,000 "words" of computer memory) is standard. With older computer operating systems (such as MS-DOS), extra memory does not make the computer significantly faster and is only a "minimum standard" that determines which programs the computer can run. With more modern computer environments (such as MS Windows), the speed of certain operations is affected by the amount of memory. Commercial CBT programs should be tested on one of the computers to be used for training to ensure that there is enough memory for the program. If software is being produced in-house, then programmers need to consider the amount of memory in the organization's computers.

3.6.1.3 Hard Disk Size

The hard disk is used to store data on the PC when the data is not being used. Most modern software programs are too large to store on floppy disks. The data stored on the hard disk can be either program, which control what the computer does, or data which the programs process. For text-based applications, a 40 Mb hard disk is usually adequate. If multimedia data such as graphics, animation, and sound is part of the CBT program, then a larger hard disk should be considered. The size of hard disk required depends on the number of programs and the type and amount of data being stored on the computer. To store large amounts of data that does not change frequently, a compact disc read-only memory (CD-ROM) drive is preferred. Note that new data cannot be stored on a CD-ROM after it is manufactured, so another data storage mechanism is necessary for updates and student data.

3.6.1.4 Computer Display Quality

The computer monitor and the video adapter card work together to display the text, graphics, and video that the PC generates. The signals from the computer go through the "video adapter" which converts digital data to the signal the monitor displays. There are several dimensions along which the adapter/monitor combination can vary:

- Resolution of video adapter - number of vertical and horizontal "picture elements" (pixels) that can be displayed; usually ranges from 320 x 140 to 1024 x 768
- Size of the monitor - measure of diagonal size of the screen: between 12" and 27"
- Number of colors - how many colors can be displayed on the screen at once; from monochrome (black and white) to 16 million colors.

The appropriate combination depends on the type of data the CBT displays. For programs that display only text, the lower resolutions are appropriate. If a program displays graphics, video, and animation, then higher-end equipment is necessary. Use the prototype CBT program to evaluate resolution requirements.

3.6.1.5 Input Devices

An input device is a computer peripheral that allows the user to enter data into the PC. The most widely known input device is the keyboard which allows the user to enter text. However, most training approaches and tasks do not require the user to enter large amounts of text. Keyboards are not widely used in the newer CBT systems since it is easier to interact with the computer through a "selection" device such as a mouse, touchscreen, or light pen. These devices allow users to interact with the graphical elements on the display through gestures such as clicking/pressing, dragging and dropping, etc.

While the mouse is the device of choice for controlling most graphical environments, touchscreens and light pens are easier for novices to use. A touchscreen is sensitive to pressure, and the coordination of computer graphics and the user's actions allow for realistic simulation of equipment and tasks. A lightpen works in much the same way, but the user touches the screen with a special pen. Although touchscreens and lightpens are easier to use than mice, they cause arm fatigue when the user is required to perform a large number of selections.

3.6.1.6 Multimedia Accessories

Multimedia can be defined as the integration of several data types in a computer program. These data types include text, pictures, animation, sound, and video. Usually, *multimedia* assumes some level of sound and video, although both do not need to be present. In the case of a PC, the presentation of sound or full-motion video requires some hardware accessories, including a sound adapter with speakers and a motion video adapter board. Currently, the most widely used method for delivering video on a computer is through a videodisc player, but digital video technology is quickly becoming widespread. A CD-ROM drive is often used to store the large amounts of data associated with multimedia. Simple text-based training programs usually do not require these accessories. However, if multimedia is needed for training, the CBT team should plan for this when choosing the computer configuration. CD-ROM players are becoming increasingly common, and several software companies are delivering programs on this media.

3.6.1.7 Network Capability

A network connection enables a PC to transfer data to other computers. This allows for data sharing between isolated computers without having to transport a diskette or CD-ROM. Data can be transferred in either direction between the students' computers and the instructor's computer. Two applications for CBT are message sending and file sharing. Two examples of message sending are for the instructor to send a note to the students or for the students' grades from a CBT program to be sent to the instructor's computer. File sharing allows for more efficient use of computer resources such as keeping a single copy of the CBT program or multimedia data on the network server instead of maintaining a copy on each PC's hard disk.

3.6.2 Examples of Standard Computer Configurations

This section describes three broad classes of PCs and is designed to assist in selecting the correct computer configuration for a specific training application. The PC classes are derived from the various levels of processing power and hardware support required to manipulate different types of information. One thing to remember when choosing a computer configuration is changing training needs. Although the first CBT programs produced by an organization may be text-based, future training systems may require more computing power. It is possible to run text-based computer programs on multimedia computers, but not vice versa, so you should plan for future needs. Table 3.6 summarizes the minimal hardware requirements for the three standard computer configurations described in this section.

Table 3.6 Minimal hardware requirements for standard computer configurations

	Text and simple graphics	**Partial Multimedia**	**Full Multimedia**
Processor	Intel 80286 processor 15 MHz clock speed	Intel 80386 processor 25 MHz clock speed	Intel 80486 processor 33 MHz clock speed
Memory	640 kb memory	1 Mb memory	4 Mb memory
Storage	40 Mb hard drive	80 Mb hard drive	150 Mb hard drive
Display	monochrome/ EGA monitor	VGA 256 color monitor	VGA 256 color monitor
Peripherals		mouse; sound board and speakers	mouse; sound board and speakers; digital video or overlay/ videodisc board; CD-ROM drive

Table 3.6 Minimal hardware requirements for standard computer configurations

3.6.2.1 Text and Simple Graphics

A text and simple graphics program has the lowest computing power requirements and can run on all first-generation IBM-PC compatible computers. Information the computer displays consists of blocks of text with some small graphics. Because such programs do not use many graphics, smaller displays with 2 or 16 colors are adequate. Also, very little computer memory or hard disk space is required because of the low storage space requirements for text and simple graphics. Multimedia peripherals are not needed. The standard configuration for this computer is

- Intel 80286 processor / 15 MHz clock speed
- 640 kb memory
- 40 Mb hard drive
- monochrome / EGA monitor.

3.6.2.2 Partial Multimedia

A partial multimedia machine adds the ability to display high-quality graphics, animation, and sound. Since this type of training applies emulations of equipment and job tasks, this level of computer hardware is required for simulation-based and microworld training. The higher data storage requirements of large graphics and sound make it necessary for the computer to have more memory and a larger hard disk. When the CBT program has significant data, it may be more cost effective to store the data on CD-ROM. The standard configuration for this computer is

- Intel 80386 processor / 25 MHz clock speed
- 1 Mb memory
- 80 Mb hard drive
- VGA 256 color monitor
- mouse or touchscreen
- sound card and speakers.

3.6.2.3 Full Multimedia

A full multimedia machine includes all the capabilities of the partial multimedia class and adds full motion video. Full multimedia is useful for showing technicians how a procedure should be carried out or for providing greater interactivity with the training. Special hardware is needed for the computer to process and display some types of multimedia data: sound cards to play sound and video cards to display full motion video. There are several computer programs that can display video without special hardware, but the video display is lower quality than that of hardware-assisted technologies. The standard configuration for this computer is

- Intel 80486 processor / 33 MHz clock speed
- 4 Mb memory
- 150 Mb hard drive
- VGA 256 color monitor
- mouse or touchscreen
- sound board and speakers
- digital video or overlay/videodisc board
- CD-ROM drive.

3.6.3 Software

Computer programs, collectively known as software, are sets of computer instructions that tell the computer hardware what to do. Without software, the computer would be useless. Software performs different functions. For example, one piece of software may control how keystrokes are interpreted, while another piece may control graphic display on the computer screen. Software is a central part of a CBT system, for it acts as a translation between the instructional designer's goals and the way the computer "acts."

There are several approaches to programming CBT; they vary in ease of use and flexibility. In most cases, the easier a program is to use, the less flexible it is. On the other hand, the more functions contained in the program, the more flexible it is. More effort is required for the user to learn to use a flexible program. Using the analogy of a car and a helicopter, it is easier to drive a car than it is to fly a helicopter, but the helicopter is more flexible with regard to the pilot's amount of control. Which one is appropriate depends on the needs, knowledge, and constraints of the traveller. Table 3.7 outlines the advantages and disadvantages of programming systems.

Table 3.7 Advantages/Disadvantages of Programming Systems

	Advantages	Disadvantages
Programming languages	• Flexible; can make the program do what is needed. • Fastest programs.	• More time needed to implement programs. • Experienced programmers needed.
Authoring tools	• Both flexible and easy to use.	• Not as flexible as programming languages; some things not possible. • Slower than programming languages.
Off-the-shelf programs	• Program is ready to run; no need to hire development team. • Low cost because it is used by many groups.	• Doesn't always match need of users.

Table 3.7 Advantages/Disadvantages of Programming Systems

3.6.3.1 Programming Languages

A programming language is a set of conventions for specifying a sequence of operations a computer performs. The operations may include displaying graphics and text on the screen, monitoring the input devices, or computating a numerical value from a formula. Like the recipe that a cook follows to transform ingredients into a meal, a computer program tells the computer hardware how to transform input (from a keyboard, mouse, or data file) into meaningful output (text display, graphics, sound, or video).

Programming languages provide the most flexibility in producing CBT systems. It is possible to produce CBT systems with programming languages that cannot be produced with authoring tools. The cost for the flexibility of programming languages is the extra effort required to produce the program. This trade- off shows up in the experience required to become an expert programmer and the high cost of hiring experienced programmers.

There are several popular programming languages, including Pascal, FORTRAN, BASIC, and COBOL. The language of choice for most programmers and most commercial programs (word processors, spread sheets, etc.) for PCs is "C." The language expertise of the programmers, the existence of any previously written software, and company rules and preferences determine which language is most appropriate for a particular project.

3.6.3.2 Authoring Tools

Like a programming language, an authoring tool is a set of conventions for specifying a sequence of operations a computer performs. The form of an authoring tool's language is different from that of a programming language. In most cases, authoring tools are much easier to use than programming languages. Many authoring tools do not require extensive experience in computer programming and require little training for users to produce working systems. Sometimes the programming is done through "direct manipulation," i.e., drawing icons or objects that control how the program will act directly on the screen. Usually, the programming language for an authoring tool (called a "scripting language") is easier to understand and use than a true programming language.

The disadvantage is that authoring tools are not as flexible as programming languages. For example, consider writing a computer program that simulates a piece of equipment. It may not be possible to do this with some authoring systems because the authoring languages place constraints on the way that the program behaves.

There are different types of authoring tools. At one end of the spectrum are general purpose systems not designed for any particular domain. These systems combine the flexibility of programming languages with the flexibility of authoring tools.

At the other end of the authoring system spectrum there are authoring tools designed for a specific domain or for a specific instructional approach. The premise is that by including domain-specific information and structure in the authoring system, it is easier to produce new training systems. These systems often support standard CBT operations such as student login, test score gathering, and grade reporting. The biggest drawback is that flexibility is lost when more structure is added. Occasionally, it is not possible to make the CBT program do what is desired.

3.6.3.3 Off-the-Shelf Programs

The simplest way to apply CBT is to find an off-the-shelf program that meets the organization's training needs. This approach requires no computer programming or instructional design since the system has been designed and implemented by another company. The cost of pre-configured training systems is often lower than that of in-house CBT since the cost is distributed across several groups. Therefore, the break- even cost for off-the-shelf CBT as compared to instructor-led training is usually in the tens of students. Comparatively, the average break-even point for self-developed CBT is considerably higher.

See Section 3.5 for a list of aviation-specific off-the-shelf CBT programs. Another source of information on off-the-shelf aviation training software is the "CBT Courseware/Hardware Matrix" report from the Aviation Industry Computer Based Training Committee (AICC, 1992).

The biggest problem with off-the-shelf programs for aviation maintenance is the small number of systems on the market. If training is needed for a specific system, chances are that CBT does not exist for that system. On the other hand, it is not necessary that training be provided for all possible systems because of similarities between systems and equipment in the aviation industry. Some off-the-shelf CBT programs for aviation maintenance training are listed in Section 3.6.5, "Example CBT Applications."

Remember: The main tradeoff between these three programming approaches is ease of use vs. flexibility. The type of CBT being created and the programmers' experience determines which one is most appropriate.

3.6.4 CBT Computer Labs

There are two basic configurations for computer training rooms: individual carrels and classroom. In the individual training configuration, workstations are arranged so that there are partitions between the individual carrels. This prevents students from disturbing each other while using the computers. A problem with this configuration is that it is difficult for students to help one another with problems. In the classroom configuration, the computers are arranged on tables that face the front of the room, and there are no barriers between workstations. This allows for instructor-led CBT and for easy collaboration among students.

The CBT laboratory should be large enough to hold the computers and computer furniture. The computer furniture should be large enough for students to place their instructional and reference manuals while using the CBT. Lighting and window shades need to be designed so that there is not excessive glare on the computer displays. Computers produce large amounts of heat, so the CBT lab should have adequate cooling.

Before designing a CBT lab and ordering equipment, the CBT team should understand the students' training needs, budget (both for installation and maintenance) for the CBT lab, and the technology issues of maintaining and upgrading computer hardware. Such issues include

- Existing computer hardware and software versions
- Compatibility of hardware and software
- Special hardware requirements to simulate equipment
- Access times, lab security, lab assistants, fee collection, etc.
- Projection equipment for instructor-led training
- Networked vs. stand-alone computers, printer access.

3.6.5 Example CBT Applications

This section describes several currently available CBT programs for aviation technicians. These programs are either designed to teach procedures for specific aircraft models or to teach general troubleshooting skills.

3.6.5.1 General Troubleshooting & Specific System Troubleshooting

These two programs produced by an aviation training CBT company typify the two main types of technical training. The general troubleshooting system is an instructor-led CBT system that provides a simulation-based training program. This three-day course covers electrical circuits, tools, and test equipment. The goal of CBT systems that teach general troubleshooting skills is to teach knowledge and skills that are applicable to many different types of systems. For example, the troubleshooting skills applied to a general aviation aircraft are the same as those for a commercial aviation aircraft, but the systems are more complex on transport category aircraft.

The second CBT is specific to a single aircraft and was developed with a Canadian air carrier. The goal of this system is to teach the user to diagnose faults in the Challenger aircraft. Although the learning in system-specific training systems may transfer to other systems, such programs usually focus on the configuration and procedures specific to that system.

3.6.5.2 Aviation Training with Video

Courseware from another aviation training company ranges from the "Principles of Troubleshooting" course that teaches general troubleshooting skills to aircraft-specific CBT for the Gulfstream II, III, and IV, Falcon 50, King Air 200, and the Citation 500 and III. Although this company has begun taking advantage of digital video in their newer systems, many of its older CBT programs are videodisc-based.

3.6.5.3 Large Aircraft Training

Another CBT company has produced "off-the-shelf" maintenance CBT for the 747, 767, MD-11, and various other aircraft. The 767 training course includes 90 hours of CBT and covers most of the major aircraft systems. These training systems include multimedia, making the system much more interesting to the user. The videodisc system includes high-quality video, still pictures, and audio.

3.6.5.4 Fundamental Troubleshooting Skills Training

A large American CBT developer is working with a German air carrier to develop courseware teaching the fundamentals of jet aircraft. Material covered in this course such as avionics, hydraulics, and mechanics is common to most aircraft and is covered in each of the type ratings. The goal is to save time by teaching this material once, instead of having to teach it for each type rating. The whole CBT program, which is 100 hours of training, is scheduled for release in July 1995.

3.6.5.5 737 and 777 Training

One commercial airline has developed three courses covering several different systems and procedures related to the 737. These courses include the APUs, tagging procedures, and fueling regulations. These are mostly one-hour courses that can be used on low-end personal computers. The airline is also working with the manufacturer on developing an instructor-led course for the 777.

3.6.5.6 ECS Intelligent Tutoring System

The Environmental Control System (ECS) Tutor was produced for the FAA Office of Aviation Medicine. This is an intelligent tutoring system (ITS) that teaches general troubleshooting skills in the context of diagnosing and repairing malfunctions of the Boeing 767's ECS. The tutor contains a deep model of the ECS which allows the user to see the consequences of his or her actions on the system down to the sensor level. The tutor is also highly graphical, allowing for direct manipulation of ECS components and realistic pictures and animation of system components and schematics.

3.6.5.7 Part 147 Training

Two companies have produced jointly several multimedia training modules for A&P mechanics. Course material consists of integrated manual and CBT software. Existing courses cover eddy current inspections, electronic communication and navigation systems, and instrumentation systems. The course material is not specific to any particular equipment or system, but teaches general principles and methods of operation for the given task. The software runs on a "partial multimedia" computer and includes the ability to track student performance.

3.7. POTENTIAL PITFALLS IN CREATING CBT

In designing and producing CBT, the CBT team needs to be aware of several potential problems that are commonly encountered. These problems may be technical, instructional, planning, ergonomic, or organizational (Table 3.8 summarizes these problems). The CBT process described in Section 3.4minimizes the chance of these problems arising.

Table 3.8 CBT Pitfalls

Problem	Prevention/Solution
Additional training not required	Perform a thorough task analysis, including an analysis of the current training methods.
Poor instructional design	Perform a thorough task analysis and an instructional design based on this analysis.
Poor user interface	Conduct frequent user acceptance evaluations during development; follow interface design guidelines and conventions; know the target users.
Poor computer programming of CBT	Apply standard software engineering and management techniques; use the right tools and programmers.
Unplanned organizational impact	Understand the shift in the instructor's role; design and evaluate a prototype course before starting school-wide use.
System not integrated with other training	Verify the whole instructional design; ensure that the students have all prerequisite knowledge; evaluate the CBT's integration with other courses.
Oversell of CBT	Set realistic goals (cost effectiveness) that the CBT should meet.

Table 3.8 CBT Pitfalls

3.7.1 Additional Training not Required

Sometimes training systems are built when there is no need for them. This usually happens when a needs analysis is not conducted. Some tasks do not require formal training because they are easy to learn, and on-the-job training is the appropriate instructional method. In other cases, instructor-led training or self- study instruction are adequate for the material being covered.

To determine if additional training is required, the instructional designer needs to perform a thorough needs analysis, including an analysis of the current training methods. The task analysis will determine if there is an opportunity for improving on-the-job performance. A needs analysis may identify ways of restructuring the task for improved performance and efficiency. If a CBT is proposed to lower training costs, then a cost/benefit analysis needs to be performed to determine if the CBT system will actually result in lower training cost.

3.7.2 Poor Instructional Design

Another pitfall in creating CBT systems is poor instructional design resulting from a mismatch between the target users' instructional needs and the training system's design. Occasionally, such a mismatch can hinder users' job performance if they are trained to perform the task incorrectly.

An important part of instructional design is making the CBT software interesting to the users. Many failed CBT programs are "page turners" that are little more than books that have been transcribed to the computer: these systems do not hold most users' attention. A program cannot be made engaging simply by tacking on pretty pictures and animation; user interest must be designed into the system from the project's beginning.

The best defense against poor instructional design is, again, a thorough job task analysis and instructional design based on this analysis. Instructional design is the bridge between the students' instructional needs and the CBT program's design. Without a thorough job task analysis, it is unlikely that the CBT program will be useful. If the program is being done in-house, iterative reviews help ensure that it will meet its instructional goals. A thorough analysis of the system's instructional design is necessary for programs built by a contractor or bought off-the-shelf.

3.7.3 Poor User Interface Design

A CBT system should be designed so that it is easy to use. The text should be readable, the controls should be intuitive, and the user should be able to determine what the program is doing and what he or she can do with the program. If this is not true, users become frustrated with the system and avoid using it. If students are forced to use a poorly designed program, they usually do not pay attention to what the program is trying to teach. A program can also be unusable and frustrating if the computer is not powerful enough for the software or does not have all of the features (such as multimedia accessories) the software requires. The net effect of poor user interface design may be a decrease in training and work motivation, so this type of failure requires special consideration.

User interface design problems can be avoided by performing frequent user acceptance evaluations during CBT development. The best way to do this is to have intended users operate the program and report any problems they have understanding its content or usage. Someone uninvolved in the CBT's programming should observe the users to record reported problems and to look for problems that users do not report. Data gathered from these evaluations should drive CBT design changes.

The CBT team should understand the target users. The team should know the users' level of computer experience, background knowledge, and learning styles. For example, if the CBT program requires a mouse and the students are unfamiliar with using a mouse, the instructor should give a short lesson on its use (possibly by letting the users play a computer game). With knowledge of user backgrounds, CBT software and hardware can be evaluated for "fit" with the students, and changes can be made, if necessary. Users have a variety of experience and knowledge, and CBT systems should be designed to be usable by the largest number of students. There is always a number of students who need extra help with any system.

3.7.4 Poor Computer Programming of CBT

A needed and properly designed CBT system will not be successful if it is poorly programmed. If the computer program does not do what it was intended to do, then it is unlikely that it will meet its instructional goals. Poor programming can result in incorrect or missing data, or programs that mysteriously "lock up" the computer. A program with too many errors is not instructionally effective because the user pays too much attention to trying to make the program work. Poor computer programming can result from several planning or management problems:

- Poor planning or resource allocation
- Wrong choice of programming tools for the task
- Insufficient training in the use of the programming tools
- Insufficient management of programming tasks, testing, or deadlines
- Personality conflicts between CBT team members.

These problems can be avoided by applying standard software engineering and management techniques to administering the programming project. Although team management is not a concern for organizations buying off-the-shelf CBT, it is still important to ensure that the training software being bought is error- free. This can be done by conducting a pilot program with a small group of students before buying the CBT program.

Because computer programming is still more of an art than a science, it is often difficult to predict how much time it will take to complete a program. On the average, it takes between 200 and 500 hours of development to produce one hour of CBT. The time for any particular project depends upon many factors, including the system's complexity, the CBT team's experience, and the amount of cooperation with management and instructors. It is important to allocate time for evaluation and testing to ensure that the CBT program is useful and usable.

3.7.5 Unplanned Organizational Impact

When applying any technology to a task, it is important to consider the impact that the technology will have on the organization. This is not often done with computer technology since it is usually assumed that the "new" method will improve efficiency and productivity. Sometimes lack of planning for the introduction of new technology results in a refusal to use the new technology. Organizational rejection of technology may be caused by poor training, confusion about the purpose of the technology, or users' feelings that the technology is being imposed on them.

When using CBT in an organization for the first time, it is important to understand and account for the shift of the instructor's role from a presenter of information to that of a student assistant. This can best be done by conducting a trial CBT course. This trial course should explicitly describe the roles of instructors, students, administrators, and support personnel. Testing a prototype in a classroom also provides useful information before the CBT course is implemented school-wide. The students should be familiar with the behavior and operation of computers and the reason for using computers in training.

3.7.6 System not Integrated with Other Training

Just as there is no single tool for all tasks, there is no single technology appropriate for all training. A CBT system should be applied to meet specific training goals for which it is appropriate. Since there are other training methods to teach other knowledge, an instructional designer must consider how to integrate CBT with other types of training. For example, a CBT system that teaches non-destructive inspection may be integrated with classroom lectures that explain the theory behind magnetic and ultrasonic NDI and textbook material that describes different NDI systems. When dividing a training course among several instructional techniques, the instructional design must be well-planned to ensure that students have all the prerequisite knowledge before proceeding to new lessons.

3.7.7 Oversell of CBT

A CBT program is an asset to technical training because it enhances traditional classroom training. CBT systems have sometimes failed because they were oversold to school management, instructors, and/or students. CBT is not the "end" in training technology; it is rather a "means" in the effort to meet the overall goal of effective and efficient technical training. The broad category of oversell includes optimistic predictions of cost effectiveness, performance improvements, development time, or user acceptance.

A good CBT system requires time, money, and other developmental resources. If sufficient resources are not allocated to building the system, it will be inadequate and will probably fail to meet its goals. If the finished CBT is ineffectively utilized, it is unlikely to be cost-effective. Another common misconception is that CBT solves all of an organization's training problems. As with any training approach, some students will have problems learning from CBT. This may result from poor CBT design or from some students' insufficient background knowledge or poor learning habits. Instructors should prepare for this eventuality.

To avoid this problem, set realistic goals for CBT. The goals should describe expected changes in training costs and effectiveness. Remember that both of these do not have to improve; in some cases, a CBT that increases effectiveness may cost more than the current training method. Finally, make sure that others in the organization understand the goals for the CBT system.

3.8 CONCLUSION

The design and implementation of CBT systems present many options to aviation training managers. While the decisions may seem complicated, they can be greatly simplified by using a proven systems approach to manage CBT selection, development, and implementation.

3.9 REFERENCES

Air Transport Association. (1991, October). Specification 104 - Guidelines for aircraft maintenance training. Washington, DC: Air Transport Association.

Eberts, R. E. & Brock, J. F. (1987). Computer-assisted and computer-managed instruction. In G. Salvendy (Ed.), *Handbook of Human Factors* (pp. 963-975). New York: John Wiley & Sons.

Flexman, R. E. & Stark, E. (1987). Design of selection and training systems. In G. Salvendy (Ed.) *Handbook of Human Factors* (pp. 1012-1038). New York: John Wiley & Sons.

Federal Aviation Regulation, §147. 14 CFR 147 (1993). Washington, DC: U.S. Government Printing Office.

Forth, K. D. (1993, March). CBT applications increasing. *Aviation Equipment Maintenance*, pp. 40-45.

Aviation Industry Computer Based Training Committee. (1992, draft). *CBT courseware/hardware matrix*.

Kearsley, G. (1982). *Costs, benefits, and productivity in training systems*. Reading, MA: Addison-Wesley Publishing Company, Inc.

Pearce, M. (in press). How much is enough? Choosing a computer-based video technology. *Proceedings of the Second Annual Conference on Multimedia in Education and Industry*.

Reigeluth, C. (1983). Instructional design: What is it and why is it? In C. Reigeluth (Ed.), *Instructional Design Theories and Models: An Overview of their Current Status*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Electric Power Research Institute. (1987, June) *Guidelines for the application of computer-based instruction* (Research Project 2294-2 Interim Report).

Woods, D. D. (1993). Process-tracing methods for the study of cognition outside of the experimental psychology laboratory. In G. A. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), *Decision making in action: Models and methods* (pp. 228-251). Norwood, NJ: Ablex Publishing Corporation.