10

CHAPTER TEN

Selecting, Testing, and Auditing IT Applications

INFORMATION TECHNOLOGY (IT) APPLICATIONS are the tools that bring value to computer systems; they drive many if not most of today's enterprise business processes. These IT applications range from the relatively simple, such as an accounts payable system to pay vendor invoices, to the highly complex, such as an enterprise resource management (ERM) set of multiple interrelated database applications to control virtually all enterprise business processes. Many IT applications today are based on vendor-leased or purchased software, an increasing number come from Web-based services, some are developed by in-house systems and programming teams, and many others are based on spreadsheet or database desktop processes. Although the IT general control procedures discussed in Chapters 6 and 7 cover controls and best practices over all IT operations, specific control processes apply to each installed IT application. In order to perform internal controls reviews in specific areas of enterprise operations, such as accounting, distribution, or engineering, IT auditors must have the skills to understand, evaluate, and test the controls over their supporting IT applications. Reviews of specific application controls often are more critical to achieving overall audit objectives than reviews of general IT controls.

Application controls, however, are very dependent on the quality of overall IT general controls. For example, if there are inadequate controls over an IT configuration management process, as discussed in Chapter 7, it will be difficult for an IT auditor to rely on the controls built into a specific application that depends on strong configuration management processes. Even though an IT auditor, for example, may find that an IT order-entry application is properly screening sales orders for valid credit approvals, the surrounding general controls must also be considered. Without IT configuration management update controls, in this example, the order-entry system's programs could be changed, without management's authorization, perhaps to override established credit approval controls.

A typical enterprise may use a large and diverse number of production IT applications. These applications support a wide variety of functions within the enterprise, starting with accounting applications but also including such areas as manufacturing, marketing, distribution, and others, depending on business activities. These supporting applications may be implemented using a variety of IT technologies, such as centralized systems with telecommunication networks, Internet-based network systems, client-server server-based applications, and even older mainframe batch-processing systems. Some of these applications may have been developed in-house but increasingly large numbers of them are based on purchased software packages installed locally or accessed through Web-based service providers. In-house-developed applications may be written in a programming language such as C# (also called C-sharp) or Visual Basic, a database report-generator language such as SQL, or the object-oriented language Java. Application documentation may range from very complete to almost nonexistent. Despite the best efforts of IT audit to suggest improvements, the same often can be said about application controls.

Even though members of management sometimes do not have a good understanding of IT general controls issues, often they are interested in IT audit issues covering specific application controls. For example, while an IT audit report finding on general controls over IT operating systems program libraries may not generate much management interest, a finding of an incorrect discount calculation based on a foreign currency conversion problem in an accounts payable application is sure to draw attention. However, because of the relative complexity of many IT applications and because their controls often reside both within the application and in supporting user areas, audits of IT applications can be challenging.

IT auditors should survey the active applications and select the more critical and appropriate ones for review. We also discuss approaches to effectively review internal accounting controls in IT applications, using several different types of applications as examples. Finally, the chapter discusses audit approaches for evaluating and testing those application controls as well as techniques for reviewing new applications under development. We focus on the internal control characteristics of different types of applications and on how to select appropriate applications in internal controls reviews. There are many differences from one application to another; this chapter focuses on how an IT auditor should select higher-risk applications as candidates for IT audit reviews, the tools and skills needed to understand and document application internal controls, and, finally, processes to test and evaluate those applications.

IT APPLICATION CONTROL ELEMENTS

People not familiar with IT sometimes think of a computer application just in terms of the system's output reports or the data displayed on terminal screens. However, every application, whether a Web-based service application, an older mainframe system, a client-server application, or an office productivity package installed on a local desktop system, has three basic components: (1) the system inputs, (2) the programs used for processing, and (3) the system outputs. Each of these has an important role in an application's internal control structure, and an IT auditor should understand these components when reviewing an IT application.

Earlier IT applications could be separated easily into these three components. As an example, the traditional computerized payroll system from long ago used time cards and a personnel paymaster file as its inputs and a set of programs to calculate pay and benefits as well as to update pay history records. The outputs from that payroll system were the printed checks, payroll register reports, and updated paymaster files. Today, that same payroll system might accept inputs from an automated plant badge reader that controls accesses and tracks attendance, a shop-floor production system that performs incentive pay calculations, various other online inputs, and a human resources database. A series of computer programs, some located at a Web-based service provider and others distributed to remote workstations, would do the processing. In many cases today, much of the payroll processing may be handled by an outside service function that does most of these activities. The modern payroll system's outputs include transactions to transmit compensation to employee bank accounts, pay vouchers mailed to employees, and input files to various tax and benefit sources, various display screens, and an updated human resources database.

Although the input, output, and computer processing system components may not be all that clear to an IT auditor performing an initial review, the same three elements exist for all applications. No matter how complex the application may appear, an IT auditor should always develop an understanding of an application by breaking down its input, output, and processing components. The next sections briefly discuss the control aspects of these application components to give an overview on selecting, auditing and testing IT applications.

Application Input Components

Every IT application needs some form of input, whether it is data manually input from transaction vouchers or supplied from some automated system. Think of a common handheld calculator: The device will generate no results unless data of some sort is input through the key panel. Although an application's programs process the data, determine the outputs, and have a major impact on controls, an IT auditor should understand the nature and sources of the input components. In traditional, batch-oriented systems, this was a fairly easy process. Application inputs often were sequential records recorded on a magnetic tape file or 80- or 90-column punched cards. Today, inputs often are generated from various automated sources, including wireless data collection devices and specialized bar code readers.

Inputs from Data Collection or Other Input Devices

Most early IT applications used punched cards as their input source. A single card carried 80 or 90 columns of alphanumeric encoded data, and users entered input transactions onto data collection sheets for keypunching onto the card formats. The original data collection sheet was the first step in the input chain, and early IT auditors were concerned that all transactions were keypunched correctly. These cards were then machine-sorted or otherwise manipulated prior to entry into a system, either read directly into a computer program or copied to magnetic tape for subsequent processing on a batch basis. That is, 500 lines of transactions may have been prepared on data collection sheets and processed as a batch. The need for all transactions to be keypunched correctly and subsequently read into the computer program made input transactions controls a key component of an application's overall internal controls.

Technology has effectively eliminated those punched-card input records today. Batch-type transactions that must be entered into an application are no longer entered by a specialized ''keypunch'' or data-entry department. Rather, operational departments use online terminals to enter their transactions for collection and subsequent processing. Following a processing schedule, these transactions may be input or collected and updated later in a batch mode. The data entry programs used to capture them often have some transaction-screening capabilities to eliminate any low-level errors common to earlier batch input systems. In many other situations, the entry of a transaction updates files in a real-time mode.

Transaction input data comes from many sources. A retail store captures sales inputs through a combination of sales entries entered on the point-of-sale (POS) terminal and product sales are entered through bar code readers. Similarly, data is captured on a manufacturing shop floor through various tickets and badges that are entered in readers by workers directly on the floor. Small computer chips—radio-frequency IDs (RFIDs)—embedded on the label of a component may provide inputs as to the product's identification and subsequent movement. All these input devices generate transactions for updating to some type of processing application. Input transactions are increasingly generated not from within the enterprise but from applications located in other physical locations and controlled by others. Enterprises today receive a wide variety of data transactions through the Internet, on older electronic data interchange (EDI) systems, or through wireless systems. In these cases, another enterprise may submit purchase order transactions, accounts payable remittances, or other significant business transactions. Individuals initiate sales transactions, trade securities, and perform other business through their home computers via the Internet. All of these represent input transactions to various IT applications, and each has its own unique control considerations.

An IT auditor reviewing application input controls always should look for some basic internal control elements that should be found in all IT applications. For example, there should be some means of checking that only correct data is entered. A computer program that, through its supporting validation tables, can verify that a product part or employee number is or is not valid cannot easily verify that the current quantity should have been entered as 100 as opposed to 10. The older batch systems had hash total checks to help check for these possible errors. A hash total is a nonmonetary value, such as the ''sum'' of all account numbers. Modern systems also need reasonableness checks built into their data collection procedures, and the programs processing the transactions need controls to prevent errors or to provide warning signals.

Application Inputs from Other Automated Systems

Today's IT applications often are highly integrated, with one application generating output data for processing by another. The transaction entered into one application may impact a variety of other interrelated applications. Thus an error or omission of an input at one point in a chain of applications may impact the processing of another connected application. In addition to understanding the sources of the transactions to an application, an IT auditor should understand the nature of other automated inputs to that same application. For example, a modern payroll system may receive inputs from a sales performance system to calculate commissions. The sales performance file that feeds the payroll system is another input. The controls there are based on the input, processing, and output controls of the sales performance system. If sales performance data represent a significant input to the payroll system, an IT auditor needs to be concerned about the controls over it as well as over any other supporting applications.

A large network of interconnected applications can present a challenge to an IT auditor attempting to review the input controls for just one application. The IT auditor may be interested in understanding application input controls for application X. However, files from applications A, B, and C may provide inputs to X while D and E provide inputs to applications A and C, respectively. An IT auditor typically does not have the time or resources to review all of these processes and must decide on the most critical ones and assume that the other less critical supporting applications are generating appropriate transactions.

Files and Database Inputs

Although usually generated by some other supporting application or updated by the application under review itself, an application's files and databases represent important inputs. In some instances, these files represent tables of data used for the validation of program data. As part of gaining an understanding of an application, an IT auditor should understand the nature and content of all supporting application files. The software that controls these files generally has various record-counting and other logical controls to determine that all transactions are correctly written onto and can be retrieved from the supporting system. Files also should have their own dating and label-checking controls to prevent them from being improperly input to a wrong processing cycle or an incorrect application. Once written as streams of sequential records on magnetic tape, today's files are input onto higher-density disk drives or USB cartridges. However, an IT auditor needs to have a general understanding not only of the type and nature of inputs to a computer application but also of the source of the file data and any controls over it. (See "Completing the IT Application Controls Audit" later in this chapter for more details.)

Databases can present particular challenges to an IT auditor. Although the term *database* often is misused to refer to almost every type of computer file, a computer system database is a method of organizing data in a format such that all important data elements point or relate to each other. In past years, many mainframe computers used what were called *hierarchical databases*, where data was organized in a grandparent-parent-child "family tree" type of structure. Using it in a manufacturing enterprise, each product might be organized as a header record that would point to each of its parts.

Those components in turn would each have a hierarchy of records comprising its individual parts. File integrity was very important here; a program error that breaks one of the connecting chains would make it difficult to retrieve the lost data.

Today, the relational database is a much more common file structure found on all types and sizes of computers. A relational database is like a multidimensional Excel spreadsheet. That is, the user can retrieve data across various database rows, columns, and pages rather than having to go to the head of each tree and search down through it to retrieve the desired data. Besides being very effective ways to organize input data to application systems, these databases allow for easy retrieval of reports for end users. Two common examples of the relational database model are Oracle Corporation's database products and IBM's DB2 database.

Application Programs

Applications are processed through a series of computer programs or sets of machine instructions. The traditional payroll application mentioned earlier would consist of computer programs to read employee's time card data, store the number of hours worked, and use the employee number on that input time card to look up the employee's rate and scheduled deductions. Based on this match, the program looks up the employee's rate of pay and multiplies this by the number of hours worked to calculate the gross pay.

A computer program is a set of instructions covering every detail of a process. A programmer writes detailed instructions for a computer system to follow. As an experiment to understand the details required to write such a larger computer program, an IT auditor who lacks programming skills should try to write down each step to follow in the morning from the time the alarm goes off until he or she arrives at the office. The next morning, the IT auditor should use these same instructions *exactly* as they are written to get up, wash and dress, and then go off to work. Following this program, most people will encounter program errors and arrive at work missing an item of clothing or worse. This is the difficulty of writing detailed computer programs. Usually it is not necessary for IT auditors to know how to write formal computer programs today beyond the simple audit retrieval applications discussed in Chapter 13, but the effective IT auditor should understand how computer programs are built and what their capabilities are in order to define appropriate control procedures.

Traditional Mainframe and Client-Server Programs

Mainframe, or what we often call legacy-type computers were used extensively for business applications since the early 1960s. These applications first were programmed in what are called first-generation actual machine languages that used binary 1s and 0s. We quickly moved to second-generation languages, what were called the assembly languages. These symbolic languages used codes to represent instructions, such as to add or store a value. Third-generation, or compiler, languages soon followed. They used actual English-like instruction statements, such as "ADD A TO B." to describe the actions to be taken. Programs called *compilers* translated these instructions into machine language. A large variety of these compiler languages were introduced in the 1960s, but COBOL[1] became the almost standard language for business data processing well into the 1980s. It is still in use today for some business applications, but specialized database and report-generator languages and object-oriented languages are now much more common.

A wide range of computer languages are used today; they include Visual Basic and Java. Many applications also are developed using English-language-like report generator languages that reside on top of a supporting computer language. Other than having the skill to write an audit retrieval request, as discussed in Chapter 13, an IT auditor today does not need to be skilled in a programming language.

Modern Computer Program Architectures

In the mainframe computer days of years past, business applications almost always were developed in-house and often were written in COBOL. Most enterprises today generally purchase or lease their software packages or access them through a Web service provider, although some IT functions still do develop their own applications. In-house development normally occurs when an enterprise has business requirements where no commercial software packages seem correct or, more significantly, when an enterprise has plans for some new strategic software-based initiative. An IT auditor today, even with a fundamental knowledge of a language such as Visual Basic, COBOL, or C, may have some initial difficulties understanding how object-oriented applications are programmed and constructed. Often these newer applications consist of many very small program code modules that pass data to one another, sometimes over remote telecommunication lines. While it is certainly not a typical IT audit need, Exhibit 10.1 describes some object-oriented high-level programming concepts. Java and C++ are two of the programming languages of today's Web-based applications.[2] An IT auditor should rely on the overall application program standards in place as well as on other programming development and maintenance controls. Rather than looking for these application programming standards in each given application reviewed, he or she should review the general systems development controls in the IT enterprise. These might be included in a general review of IT operations, as discussed in Chapter 6.

When an enterprise plans to build and launch in-house a major new or revised software application, IT audit should request the right to perform a preimplementation review of the new application development project.

Preimplementation IT audit reviews are most effective for large development efforts that cover an extended span of time and primarily components developed in-house. Exhibit 10.2 contains IT audit procedures for a review of a new application systems development controls. These control processes are closely linked with the IT general controls discussed in Chapter 6 and an IT auditor should look for them in each application selected for review. Today many new application development projects do not consist just of in-house-developed new programs. Many modern applications are constructed by building data reference tables as part of purchased software applications as well as building interfaces between these purchased applications and other existing components. Proper attention must be devoted to preserving internal controls and performing adequate testing in these situations, and the IT audit preimplementation review approach can provide service to the enterprise.

Object-oriented programming (OOP) programming languages, such as JAVA or C++, model organized around ''objects'' of data rather than logic-based ''actions.'' Programs using languages, such as COBOL, were based on logical procedures that took input data, processed it, and produced output data. These older programming approaches described the processing logic but did not define the data. Object-oriented programming focuses on the data objects we want to manipulate rather than the logic required to manipulate them. Examples of objects range from human beings (described by name, address, etc.) to buildings and floors (whose properties can be described and managed) or the individual parts in a manufactured product.

The first step in OOP is to go through a data modeling exercise and identify all the objects you want to manipulate and how they relate to each other. Think of all of the furniture in the board of directors' meeting room. There will be a major table for board meetings and side tables for the support staff. The chairs in that room will be *objects* with each director, around the table, having one *class* of chair. The support staff, another class, and the CEO at the end of the table with still another. These objects are then generalized into *classes of objects*. OOP defines the logical sequences of these classes of objects. The directors' chairs are arranged around the conference table, the CEO at the end, and support staff off to the sides. OOP provides computer instructions, based on the relevant data in the class object characteristics, to allow the objects and their characteristics to communicate with each other in well-defined interfaces called *messages*. For example, the CEO's chair will be at the head or the table, and if the CEO is present, messages will be delivered to other board members.

The concepts and rules used in object-oriented programming provide these important benefits:

- The concept of a data class makes it possible to define subclasses of data objects that share some or all of the main class characteristics. Called *inheritance*, this property of OOP forces a more thorough data analysis, reduces development time, and ensures more accurate coding.
- Since a class defines only the data it needs to be concerned with, when an instance of that class (an object) is run, the OOP program will not be able to access other program data accidentally. This characteristic of *data hiding* provides greater system security and avoids unintended data corruption.
- The definition of a class is reusable both by the program for which it is initially created and also by other object-oriented programs (and, for this reason, can be more easily distributed for use in networks).
- The concept of data classes allows a programmer to create new *data types* that is not already defined in the language itself.

The OOP languages C++ and JAVA are perhaps the most popular object-oriented languages today, with JAVA frequently used in distributed applications on corporate networks and the Internet.

EXHIBIT 10.1 Object-Oriented Programming Language Concepts

Vendor-Supplied Software

Today most IT applications are based on vendor-supplied software. An outside vendor will supply the basic, often Web-based, system elements, and the enterprise's IT development function is responsible only for building custom tables, file interfaces, and output report formats around the purchased or licensed application. Often the vendor protects the actual program source code for the purchased software to prevent improper access and changes. IT auditors should be concerned that the software vendor has a reputation for quality, error-free software. Often smaller, entrepreneurial software suppliers offer very cost-effective solutions, but there can be risks in using undercapitalized software developers. If there is any doubt about the software vendor's

1. All requests for new or revised applications should follow IT standards and receive prior authorization.
2. The application development process should include sufficient requesting user interviews to develop a firm understanding of needs.
3. All new application projects should receive a detailed statement of requirements along with a formal cost benefit analysis.
4. Project plans should be prepared for all IT department development work as well as for individual application development projects.
5. Care should be given to ensuring that application development projects meet the long-range objectives of the enterprise.
6. The responsibilities for applications development work should be assigned with adequate time allowed to complete development assignments.
7. The applications development process should include sufficient user interviews to obtain a full understanding of requirements.
8. Attention must be given to internal controls, audit trails, and continuity procedures.
9. Adequate resource and capacity planning should be in place to ensure that all hardware and software is sufficient when the application is placed in production.
10. Sufficient attention, including test procedures, must be paid to backup, storage, and continuity planning for the new application.
11. Adequate controls must be installed to provide strong assurances regarding the integrity of the data processes and outputs from the application.
12. The application should be built with adequate controls for the identification and correction of processing errors.
13. All application processing data and transactions should contain strong audit trails.
14. Adequate documentation should be prepared on a technical as well as an application user level.
15. Test data should be prepared following a predetermined test plan that outlines expected results and satisfies user expectations.
16. When data is converted from an existing application, strong control procedures should be established over the conversion process.
17. If a critical application, internal audit should be given an opportunity to participate in a formal preimplementation audit.
18. There should be a formal sign-off and approval process as part of the completion of the application development process.

EXHIBIT 10.2 Internal Audit IT Application Development Review Guidelines

stability, arrangements should be made at the time of the software purchase contract to place a version of the vendor's source code in escrow in the event of its business failure. A bank or some other agency would hold a version of the protected source code for release to customers if the software vendor were to fail.

The decision to license, lease, or purchase a software package too often is based on an IT manager meeting a software salesperson at a trade show, establishing a need, and acquiring the software package without a full analysis of its costs and benefits. While lacking any form of the traditional IT preimplementation review, IT auditors can play a strong consulting-level role supporting IT management in the acquisition of a new software package. There are often many internal control issues that should be considered beyond descriptions in vendor sales brochures. Exhibit 10.3 presents an IT audit review procedure to use both when providing consulting help and when reviewing

	Audit Internal Controls Review Procedure	Workpaper Reference
1.	Determine that the requirements and objectives for the new application have been clearly approved and defined.	______
2.	Assess whether application requirements have been clearly defined and whether they can be satisfied by modifications of current application.	______
3.	If requirements call for a new application, determine whether an IT analysis has been performed to determine if it may be more cost-effective to develop in-house or to purchase.	______
4.	If a search for a potential purchased application must be undertaken, determine that detailed requirements have been defined through a request for proposal (RFP) approach.	______
5.	Determine that the RFP requirements for the application clearly match the existing enterprise IT environment.	______
6.	Review procedures for the distribution of RFPs to ensure that this distribution covered all appropriate vendor candidates, and raise questions if a known vendor appears missing.	______
7.	Assess whether IT enterprise documentation is in place to review all vendor proposals on a consistent basis.	______
8.	For application software vendors that appear to meet preliminary requirements, determine that the software has been effectively demonstrated through appropriate testing.	______
9.	Where multiple vendors are presenting competing software products, consistent evaluation procedures should be in place.	______
10.	Enterprise financial and legal resources should be in place to participate in software selection.	______
11.	Determine that the selected software product has adequate documentation, help facilities, and a regular update program in place.	______
12.	Determine that an implementation plan is in place to convert either data or an existing application to the new software application.	______
13.	Where appropriate, develop preliminary plans for CAATT procedures covering the new application.	______
14.	Establish internal audit workpaper records for the new, purchased software application.	______

EXHIBIT 10.3 Purchased Software Internal Controls IT Audit Review Procedure

the decision to purchase a major new software package. An IT auditor should understand the internal controls of major purchased software applications as well as he or she understands any in-house-developed application.

Large, integrated packages, such as ERP systems, can have a major impact on all aspects of an enterprise. These database application packages may include production, purchasing, inventory, human resources, accounting, and all other business applications implemented as a linked series of databases. Data introduced to one application component, such as a revised standard cost for a manufactured part, will connect to other connected systems as necessary. For example, that revised standard cost will be reflected in inventory and financial systems, among others.

IT Application Output Components

No discussion of an application system would be complete without a description of its output components. These key application components usually consist of output screens, updated files, or even printed reports. These are important areas to survey in any application review, and IT audit should be concerned about controls contained on the output screens and control files. Older applications produced large volumes of output reports indicating the results of the processing and any control or error problems. The sheer volume and frequency of those reports often prevented users from paying adequate attention to control problems, and IT auditors frequently found control concerns that users could have identified just by reviewing their output reports.

Today's applications produce far fewer (if any) paper-based output reports; instead, results are reported on online data retrieval screens. In some cases, special online reports signal control problems and data errors; in others, users are responsible for calling up the appropriate screen to review problems. All too often, users ignore this step, and processing errors can go undetected. IT auditors always should review the scope of application output reports, screen messages and their user dispositions. Reports or screens are not the only application outputs. Transactions or updated files typically are passed to a variety of other integrated applications. Just as a modern IT application may receive inputs from a highly integrated set of input systems, it may be one link in a chain to other applications. Again and always, an IT auditor should develop a good understanding of the application reviewed as well as all of its inputs and outputs.

SELECTING APPLICATIONS FOR IT AUDIT REVIEWS

Although all significant IT operations and key applications should be subject to regular internal control reviews, IT audit typically does not have the resources or time to regularly review the controls for all enterprise IT applications. In addition, many IT applications represent a minimal level of control risk and are not essential audit candidates. As part of a specific operational review or a general IT controls review, IT audit should select only its more critical applications for review.

An IT auditor should use the risk assessment techniques discussed in Chapter 4 to identify the more critical application vulnerabilities pertaining to an enterprise's reporting, operational, and compliance requirements. Based on a high-level understanding of potential application review candidates, an effective approach is to rate all applications on a scale from 1 to 5 for each category according to these criteria:

- Does the application contain primary enterprise controls or functions?
- Based on IT audit's preliminary review, what is the design effectiveness of the application's internal controls?
- Is the application primarily based on vendor-supplied, prepackaged software, or was it developed in-house?
- Does the application support more than one critical business process?
- Has this application been changed frequently or is it stable from period to period?

- What is the complexity of making application changes (e.g., table changes versus program code changes)?
- What is the financial impact of the application controls?
- What is the overall effectiveness of the IT general controls that support the application (e.g., change management, logical security, and operational controls)?

Each application reviewed should be scored with higher numbers representing a create-audit criticality concern. For example, if application changes require updating program code, the score might be 5; changes through external tables might receive a 1 or 2. These scores, of course, are very arbitrary, but they provide a measure for each application considered.

Exhibit 10.4 shows an example of an application's control risk assessment using these scoring criteria. We have listed three example applications, A, B, and C, and have displayed them on a spreadsheet form with sample scorings. We have assigned a weighting for each of these criteria. For example, column 1 on application primary controls is given a weighting of 20 while column 6 on the complexity of changes is given a 10. Based on individual scores and their weights, a criticality score can be computed for each application. For example, for application A, this would begin as $(5 \times 20) + (1 \times 10) + (5 \times 10) + \ldots = 375$. The application that has the highest criticality scores would be the most appropriate, higher risk applications for planning an IT audit.

Although a criticality score "wins" as a candidate for application review, IT audit also must consider other factors when selecting IT applications for review. Because IT applications are often so critical to enterprise operations, IT auditors often receive specific requests from their audit committee or management to review specific application controls. Some of the factors that may further impact IT audit's decision to select one specific application over another may include:

- **Management requests.** Management often asks IT audit to review the controls in newly installed or other significant IT applications due to reported problems or their perceived strategic importance to the enterprise. Sometimes these management requests are not made for the correct reasons, however. For example, sales analysis reports may appear to be incorrect due to bad data submitted from a reporting division, but management may consider the incorrect reports to be a "computer problem" and request an IT audit application review. Initially IT audit may not be aware of such user input problems and would perform normal review procedures. When IT audit is aware of such mitigating circumstances, audit test strategies should be modified prior to starting the review.
- **Preimplementation reviews of new applications.** In many instances, IT audit should become involved in reviewing new applications before they are placed in production. This is true for in-house-developed applications and purchased software packages. Strategies for IT audit preimplementation reviews are discussed in the section titled "Application Review Case Study: Client-Server Budgeting System" later in this chapter.
- **Postimplementation application reviews.** For some critical applications that are subject to a risk analysis, IT auditors also may want to perform a detailed

	Application Selection Risk Factor Assessment								
	20	10	10	10	10	10	15	15	
Application	Application contains primary controls or functions	Design effectiveness of the application	Vendor supplied, prepackaged, or in-house developed	Supports more than one critical business process	Frequency of application changes	Complexity of application changes	Application financial impact	Effectiveness of IT general controls support	Composite score
Application A	5	1	5	5	3	3	5	2	275
Application B	1	1	2	1	1	1	4	2	170
Application C	5	2	2	1	5	5	5	2	245

EXHIBIT 10.4 Application Control Risk Assessment Example

application review sometime shortly after the actual system implementation. If an application has sufficient financial and operational controls significance, IT audit may want to schedule at least limited controls reviews on an ongoing basis.

- **Internal control assessment considerations.** Chapter 1 discussed the need for evaluating and testing internal controls as part of the Sarbanes-Oxley (SOx) Section 404 internal controls review process, and an application controls assessment is an important part of that evaluation. The results of understanding, documenting, and testing specific IT application controls by IT audit may provide a basis for the external auditor reviews in their SOx attestation processes.

IT audit typically is faced with requests for reviews of a large number of application candidates at any time. Auditors must take care to document why one application is select over another.

IT auditors often perform reviews of the specific applications that support an overall functional area. For example, the enterprise IT audit may schedule a combined operational and financial review of the purchasing department. This also may be the appropriate time to review the application controls for the major automated purchasing systems supporting that department. In this integrated audit approach, IT auditors can concentrate on the more technical issues surrounding the applications and on other supporting operational controls.

PERFORMING AN APPLICATION CONTROLS REVIEW: PRELIMINARY STEPS

Once an application has been selected for review, IT audit should gain a more detailed understanding of the purpose or objectives of that application, the technology approaches used, and the relationship of the application to other related processes. It may be necessary for the assigned IT auditor to do some background reading and study special technical aspects of that application. Auditors often can acquire this knowledge by reviewing past audit workpapers and applications documentation and by interviewing IT and user personnel. As an early step in this review process, IT audit should perform a walk-through of the application to better understand how it works and how its controls function. These preliminary steps will allow an IT auditor to develop specific audit tests of the application's more significant controls.

In the early days of enterprise-developed IT applications, documentation often consisted of detailed system flowcharts with supporting record layouts and little else. This documentation helped the programmer but usually was of little use to application users or IT auditors attempting to understand the application's controls. In addition, these early flowcharts were often hand-prepared and became out of date quickly. When one relatively small change was made to a complex system flowchart, designers often were reluctant to redraw their charts. Although they might have remembered the changes, other persons reviewing the documentation would not be aware of them.

Over time, application documentation evolved into a format that was oriented more toward text and functional charts. Decision tables and logic charts described the functions of individual programs while text described the overall system. Although this type of documentation was more functional and less technical, it also quickly became outdated. Programmers and system designers often did not incorporate changes into this systems documentation. Today, powerful documentation tools, such as flowchart generators, are available. A real strength of these automated documentation tools is that detailed flowcharts can be combined into summary versions with changes introduced on one chart updating all others.

IT auditors can expect to find various types and amounts of application documentation depending on the relative age and complexity of the application. Applications developed in-house sometimes have very limited documentation; in contrast, vendor-supplied applications that often have extensive documentation, including many dozens of volumes of descriptive text. Users will sometimes treat such documentation as almost encyclopedic reference materials. A review of the published documentation should be a first step to gaining an audit understanding of an application. If aspects of the documentation are missing or out of date, the IT auditor probably will have an audit report finding at the conclusion of the review. However, lack of documentation should not necessarily prevent an IT auditor from performing an application review. When performing the review, IT audit normally should look for these documentation elements:

- **Systems development methodology (SDM) initiating documents.** These documents include initial project requests, any cost/benefit justifications, and the general systems design requirements. Although many initial assumptions may have changed during the systems design and implementation process, these documents often help IT audit understand why the application was designed and controlled in the manner it is.
- **Functional design specifications.** This documentation should describe the application in some detail, including each of the program elements, database specifications, and systems controls. If major changes have been made to the application since its original implementation, the design documentation should reflect these changes. The purpose of this documentation is to allow an IT analyst to make changes or respond to user questions regarding the application.
- **Application and program change histories.** There should be some type of log or documented record listing all program changes within an application. Some IT departments keep such a log with the application documentation; others maintain it in a central file cross-referenced to the program source code. This type of documentation is an essential element to control program changes; it also provides IT audit with some feeling for the application's relative stability. A large number of ongoing change requests for a given application may mean that the application system is not achieving user objectives. Revision service support controls should follow information technology infrastructure library (ITIL) service support best practices as discussed in Chapter 7.
- **User documentation manuals.** Along with technical documentation, appropriate user documentation should be available for any application. In a modern, Web-based system, much of this user documentation may be in the form of "HELP" or "READ ME" online screens. However, this documentation should be sufficiently

comprehensive to answer a wide range of user questions. It also should be supported by evidence of a user training program, as appropriate.

IT audit should review selected application documentation to gain an understanding of the controls to be reviewed and perhaps use these materials to develop questions for later audit review interviews. Auditors also should take copies of key or representative sections for workpaper documentation. However, normally an IT auditor should not attempt to copy the entire documentation file for workpaper purposes. Many times auditors copy too much, adding considerable bulk to workpaper files but does little to accomplish audit objectives.

Conducting an Application Walk-Through Review

Once IT audit has reviewed prior workpapers, the application's documentation, and interviewed both users and IT personnel to clarify any questions raised by the documentation review, a next step is to verify the application controls and processes by a *walk-through review*. For an IT application, a walk-through review is similar to an initial review of an operational facility where the auditor tours a facility, such as a production floor. The purpose of this walk-through is to confirm IT audit's general understanding about how the IT application operates. During the walk-through, the auditor preliminarily tests application controls through sample transactions.

As an example of an application walk-through review process, assume that IT audit has been asked to review the controls over an older in-house-developed accounts payable application operating on an in-house server system. The enterprise is a manufacturing firm with other fairly sophisticated IT applications. Assume that this accounts payable application was installed several years earlier and was never reviewed when it was under development. Now management has asked IT audit to review the application's internal controls due to integrity concerns. Based on the review of application documentation, IT audit has determined that the application receives inputs from these sources:

- Purchase order commitments from the manufacturing material requirements planning purchasing system
- Notifications of goods received from the materials-receiving system
- Various online terminal payment transactions for indirect goods and services that are not recorded through the materials-receiving system
- Payment approval transactions entered through an input screen
- Miscellaneous payables journal transactions entered as batch data

Application data is recorded on a relational database along with tables of values for validating purchase terms, including the calculation of any purchase discounts. Based on the documentation review, application outputs include the accounts payable electronic fund transfer transactions, paper checks, transactions to the general ledger and to cost accounting applications, and various control and accounting summary screens and reports.

The prime system users here are personnel from the general accounting and purchasing departments, who set up automatic vendor payments under preagreed

1. Develop a general understanding of the application, its inputs and outputs, and any procedures requiring manual or other system interactions.
2. For an application with a large number of steps requiring manual processing procedures, select a sample of key transaction types to be processed from a normal production cycle. For workpaper documentation purposes, document identifying control numbers or other characteristics to trace transactions through application processes.
3. Observe or use software tools to monitor the processing of each module or workstation step, noting situations where the walk-through transaction has:
   a. Inputs to another application or supporting process are passed on through the node processing module.
   b. Transactions are held for further cycles in process or rejected as errors during the specific processing module.
4. Follow selected transactions through each processing module step, documenting instances where the documented control procedures are not being followed or where the transaction causes application errors or manual operator difficulties.
5. At the end of the walk-through, discuss with appropriate IT or user administrators any unusual or unexpected problems and document internal control status.
6. For an automated application with essentially no paper trail, follow essentially the same procedures but make appropriate inquiries and use software query tools to determine if the application is processing with appropriate controls and as expected.

EXHIBIT 10.5 Application Walk-Through IT Audit Review Steps

terms. The example application flowchart in Exhibit 10.5 describes general IT audit steps for an application walk-through review, where P/O stands for purchase order and the other key document is the purchase requisition. The steps to performing an application walk-through for the example accounts payable application include:

1. **Briefly describe the application in the audit workpapers.** Based on its review of the documentation, IT audit should prepare a brief description of the application for later inclusion in the audit workpapers. This workpaper documentation follows the general format of the walk-through description except it provides greater detail, identifying key subsystems, input screen formats, key data file names, and output report formats. (For a discussion of IT audit workpapers, see Chapter 5.)
2. **Develop a block diagram description of the application.** A block diagram represents an abbreviated auditor-level systems or functional-level flowchart for the application. It should reflect the description cited in step 1 and also illustrate some application flow concepts. Numerous laptop system software tools are available to create the diagram; or a flowchart can even be a hand-drawn document to increase an auditor's understanding of the application reviewed. Exhibit 10.6 is an example of such an application process block diagram that an IT auditor can use to confirm his or her understanding of the application with key IT and user personnel.
3. **Select key application transactions.** Based on the previous steps, the IT auditor should select one or more representative transactions to trace through the application. This selection would be based on discussions with users and fellow members of the internal audit team. In this example of an accounts payable system, the IT auditor may select automated transactions that the receiving system should match against the payables purchase order records to initiate payment.

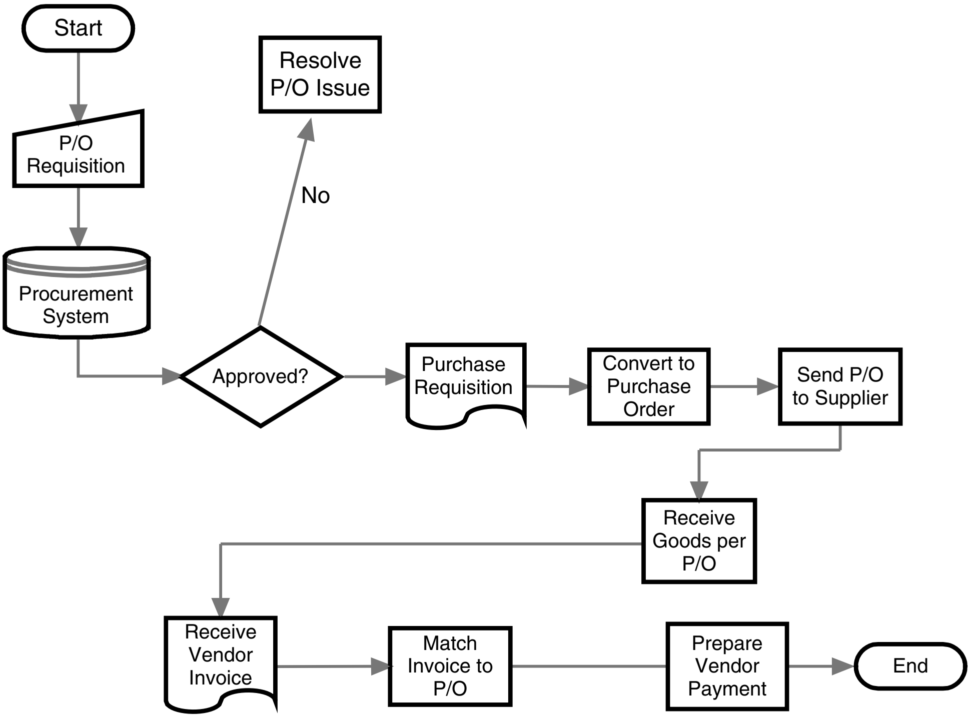

EXHIBIT 10.6 Application Process Block Diagram

4. **Walk a selected transaction through the system processes.** In the older days of manual or simpler IT applications, a walk-through amounted to just that. That is, an auditor would take an input transaction form and walk it through each of the clerical desks or steps normally used to process the transaction in order to verify the processing procedures. In a modern application, this walk-through process typically requires recording "screen shot" prints of a transaction as it is entered into a terminal and then prints that follow the transaction through subsequent steps. In our example, the walk-through transaction is a receiving report entry indicating that a valid open purchase commitment had been received. IT audit then would review the open commitments module of the system to determine whether the transaction was recorded on a transaction report or screen, then trace the transaction to a properly computed accounts payable check or to a funds-transfer transaction and then to general ledger system transactions for the correct amount.

   This type of application testing is called *compliance testing*. That is, IT audit is verifying that the application is operating in compliance with preestablished control procedures. *Substantive testing* or a test of financial statement balances involves a comparison of account balances or other methods; this is used if IT audit wishes to verify that all accounts payable checks have been input to the general ledger. Tests in support of SOx Section 404 controls typically tie a single-item test to financial statement general ledger accounts.

5. **Modify the system understanding as required.** The purpose of an application walk-through is to develop a basic understanding of the application's functions and controls; thus, a walk-through review does not allow IT audit to determine whether *all* transactions are working as described. However, if IT audit discovers that the selected walk-through transactions are not working as assumed, the preliminary auditor-prepared application documentation may need to be revised. Once that documentation has been revised, IT audit may want to repeat the previous steps to determine that the auditor has a proper understanding of system transaction flows.

Walk-throughs allow IT audit to gain a preliminary understanding not only of the application and its controls, but also of its relationship with other automated systems. Limited compliance testing allows the IT auditor to confirm that the application is operating as described. Although it is not a substitute for detailed, substantive application testing, a walk-through allows an IT auditor to identify major internal control weaknesses and gain a sufficient understanding of the application to define control objectives for subsequent, detailed audit testing and evaluation procedures.

Developing Application Control Objectives

After the review of documentation and walk-through compliance testing, an IT auditor should develop detailed audit objectives and procedures for completing the application review. This definition of audit objectives depends on the type of review planned, the characteristics of the application, and the results of the preliminary review steps. A particular review might be concerned with the level of control risk and the ability of the application to support financial statements correctly. The procedures associated with these audit objectives would be tests of the financial statement balances built up from detailed application transactions.

An IT auditor also could have other objectives in reviewing an application. Management may have asked IT audit to review an application to determine if users have received sufficient training to operate it or to determine if related discount and interest calculations associated with accounts payable are performed correctly. The walk-through compliance testing may have identified significant problems, and an IT auditor may want to do little more than confirm those preliminary but troubling observations.

Specific application review audit objectives should be clearly defined. An IT auditor responsible for the detailed review might wish to summarize these objectives for appropriate members of management to review and approve. Doing this may help prevent an IT auditor from devoting resources to testing an area not considered significant. In the above-mentioned accounts payable system, an IT auditor may have established several specific objectives for this review:

- The accounts payable system should have adequate internal controls, such that all receipts recorded from the receiving system are correctly matched to vendor files before the preparation of disbursements.
- Vendor terms should be computed correctly with controls to eliminate potential duplicate payments.

- Controls should be in place to prevent or at least flag improper or unusual disbursements.
- All systems-generated disbursements should be recorded on general ledger files using correct account numbers and other descriptive codes.

Depending on management's direction, IT audit might develop other objectives for performing such a review. For example, the review could focus on database integrity or on control procedures over miscellaneous disbursements. Any review may have multiple objectives. For example, if management had asked IT audit to review the accounts payable system to ensure that no illegal or improper payments have been made, IT audit probably would also want to add a general objective to assess control risk and to determine that the system of internal controls is adequate.

Before starting any detailed application review, IT audit should document the specific objectives of the review and discuss them with those managers who had requested the review to determine if the planned approach is on target and will satisfy the audit request. This same procedure should take place even if the application review has been initiated by the internal audit department as part of a total review of an IT function. Exhibit 10.7 contains some suggested control procedures and audit objectives for an IT application

1. Develop a general understanding of the application to be reviewed: its principal business purposes, inputs, outputs, and technology environment.
2. Based on the general understanding of the application, develop a general process flowchart that identifies key decision points, inputs, outputs, and internal controls.
3. Develop an understanding of the general controls surrounding the application and its processing environment, with an emphasis on ITIL service support and service delivery general controls. (See Chapter 7.)
4. Discuss the application and its performance with key system users and IT to understand any concerns or outstanding issues.
5. Develop a testing plan for the application that emphasizes:
   a. Identification of significant transactions, accounting, and business-related controls within and surrounding the application's environment.
   b. Identify control objectives covering each of those significant controls as well as areas of concern that should satisfy the auditor that key controls are effective.
   c. Develop testing and sampling approaches for each of the key controls.
6. Gather evidence to perform tests of identified key controls, including:
   a. Copies of key files and extracts of transactions to reperform application functions.
   b. Special application transactions to test key or critical application controls.
   c. CAATT procedures or package software functions, if appropriate, to review application transactions and special functions.
   d. Manual or paper-based documentation to support the applications controls testing.
7. Schedule and perform tests of key application controls using the gathered test materials.
8. Evaluate all test results using a pass/fail approach, and communicate testing results with key systems users and IT to verify and validate the testing approach and its results.
9. Maintain copies of all testing plans and evidence, documenting the results in internal audit workpapers.
10. Develop an appropriate corrective action plan, where appropriate, to correct any problems encountered in the testing or application review.

EXHIBIT 10.7 IT Application Review Control Objectives and Audit Procedures

review. Because there are so many types and variations of IT applications, the exhibit only lists high-level audit review approaches, and an IT auditor should make other changes as appropriate.

COMPLETING THE IT APPLICATION CONTROLS AUDIT

Usually more difficult for an IT auditor to define than the objectives for an IT audit of general controls, the specific audit objectives supporting a detailed IT application audit can vary depending on whether the review covers a single application or is a module of a larger business process, such as an ERP system. The IT auditor's review strategy depends on whether (1) the application primarily uses purchased or in-house-developed software components; (2) the application is integrated with others or is a separate process; (3) it uses Web-based service providers, client-server or older, legacy computer system methods; and (4) its controls are largely automated or require extensive human intervention actions. The exact nature of an application also can vary considerably. Although the emphasis of audits once primarily was over controls in accounting-related applications, IT auditors today may review applications implemented in other areas as well such as manufacturing production planning or loan portfolio analysis. Any of these areas requires knowledge of the application's specific attributes as well as the supporting technologies. That is, an IT auditor should understand and document how the application works, then define specific audit test objectives, and finally perform a series of audit tests to verify that the application controls are in place and working as expected.

Besides the review of documentation and the walk-through, discussions with key user and responsible systems personnel can aid in the IT auditor's understanding. The amount of effort spent here depends both on the type of application reviewed and the number of users who can be of help. For example, a capital budgeting decision support application probably will have a small group of key users who have a thorough understanding of its procedures. A logistical support system, such as factory floor data collection, may be used by a large group, where it may be difficult to identify just the key system users.

The next step is to complete the documentation of the application for audit purposes. IT audit should have been making workpaper notes throughout. The documentation procedure here is largely one of summarizing where workpapers describe the understanding gained and include notes for potential follow-up review work.

Clarifying and Testing Audit Control Objectives

The previous section discussed the importance of establishing test objectives as part of an application review—the types of controls an IT auditor would expect to be in place for an application. IT audit sometimes can fail in this important next step of defining the specific objectives of the review. For example, management may expect IT audit to review an application's internal accounting controls, but an IT audit may emphasize logical security controls with minimal attention given to other established control objectives. This misunderstanding of IT audit objectives becomes especially critical when the review is not in an IT auditor's more common realm of or related accounting or related applications. For example, if management has asked IT audit to review a new manufacturing resource planning system, its objectives could include validating internal accounting controls, reviewing for materials parts flow efficiencies, checking for system compliance with applicable regulations, or a combination of these. These objectives should be summarized briefly and discussed with both audit management and application-user management.

Although the need for a clear statement of review objectives may appear an obvious early step, IT auditors too often omit this important step. Of course, the objectives of an application review may change if IT audit encounters evidence of other control problems during the course of a review. In a manufacturing resource planning review, for example, the initial objective of affirming the adequacy of the MRP application's internal controls might change to one of fraud detection if potentially invalid transactions were encountered.

Having defined these objectives, IT audit should next test the key control points within the application. Because, as part of gaining an understanding, the auditor already has done limited compliance testing and through the walk-through, these test procedures can now be expanded to make a more definitive assessment of the application's controls. In older and simpler batch-oriented systems, this task was fairly easy. IT audit looked for input data acceptance controls, for any computer-processing decision points, and for output data verification controls. Since there are only a few processes associated with such an older batch application, this identification of test procedures often could be accomplished with minimal analysis. Modern applications today with online updating, close integration with other applications, and sophisticated programming techniques all combine to make identifying test procedures difficult. Other factors include:

- Inputs to the application may have been generated by external sources, such as Web linkages, or from other applications at partner enterprises.
- Controls once performed by data input personnel are now usually built into programs.
- Modern optical scanning input devices and output documents with multidimensional bar codes make visual inspection difficult.
- Database files may be shared with other applications, making it difficult to determine where a change or transaction originated.
- The application may make extensive use of Web interfaces and will appear to be paperless to IT audit.

There are numerous other reasons why an IT auditor may have difficulty initially identifying IT application audit test procedures. The application's description, along with key user discussions, should help to identify some of these controls. As a rule of thumb, an IT auditor should look for points where system logic or control decisions are made within an application and then develop test procedures to verify that those decision points are correct. These points represent the key controls within an application, such as checks on the completeness of transactions or the accuracy of calculations. Exhibit 10.8 lists some typical areas of audit concern, control objectives, and test procedures for a review of a client-server application.

The following are test procedures that an IT auditor might use when assessing application internal controls. Based on the nature and objectives of the application, these may not apply to all applications, and the auditor will need to have a detailed understanding of the application and its key internal controls before developing a test plan.

Audit Concern	Control	Potential IT Audit Tests
1. File data input validation	Files for processing are available and complete.	Review validation process procedures and test operations.
2. Process functionality and calculations	Specific calculations conducted on one or more inputs and stored data elements produce appropriate data elements. Existing data tables or master file rating tables should be used to validate data.	Compare input values and output values for all scenarios by walk-throughs or reperformance tests. Review table maintenance and controls, and assess adequacy of change edits and tolerance level controls.
3. Correct functionality and calculations	Automated tracking of changes made to data, associating the change with a specific user. Automated tracking and highlighting of overrides to normal processes.	Review a sample of reports for evidence of appropriate reviews. Review access to override normal processes.
4. Data extraction, filtering, and reporting	Extract routine outputs are assessed for reasonableness and completeness. Evaluation of data used to perform estimation for financial reporting purposes.	Review design of an extract routine against data files used. Review supervisory assessment of output from extract routine for evidence of regular review and challenges. Review sample of allocations for appropriateness. Review process to assess extracted data for completeness and validity.
5. Process to process balancing	Automated checking of data received from feeder systems into or ledger systems to assure validity. Automated checking that balances on both systems match: or if not, an exception report is generated and used.	Inspect interface error reports, and search for evidence of appropriate correctibe actions. Inspect validity and completeness application parameters and settings. Review access to set and amend configurable parameters on interfaces. Inspect evidence of matching reports, checks, and error file processing to determine whether controls are operating effectively.
6. Automated functionality and aging	File extracts from supporting applications are available to provide management with data on aged transactions.	Test sample of listing transactions to validate appropriateness of aging processing.
7. Checks for duplicate transactions	Comparison of individual transactions to previously recorded transactions to match fields. Comparison of individual files to expected dates, times, sizes, and other values.	Review access to set and amend configurable parameters on duplicate transactions or files. Review processes for handling rejected files or transactions.

EXHIBIT 10.8 IT Application Review Processing Controls

This exhibit some outlines areas of IT auditor concern, potential control objectives to support those areas, and test procedures to determine if the controls are operating. These steps are oriented to an almost generic financial application that IT auditors frequently encounter. However, because there are so many different types and variations of IT applications, an IT auditor will have to custom build a review approach with consideration given to these issues:

- Tests of application inputs and outputs
- Test transaction evaluation approaches
- Other application review techniques

Tests of Application Inputs and Outputs

In the very early days of IT auditing, many audit-related tests were little more than checks to verify that all inputs to a program were accounted for correctly and that the correct number of output transactions were produced based on these inputs. An auditor's review of an automated payroll system is an example of such a set of tests of inputs and outputs. The IT auditor, of days past, would perform tests to determine that all time cards input were either accepted or rejected and that the number of output payroll checks produced could be reconciled to those system input time cards. This was a very basic test of system inputs and outputs.

Although automated applications have become much more complex, many audit test procedures today are little more than those same tests of inputs and outputs. An IT auditor should examine the outputs generated from an application, such as invoices produced by a billing system, to determine that the input data and automated computations are correct. This type of audit test is limited in nature and often will not cover all transactions or functions within an application.

The purpose of a control risk assessment or compliance test is to determine if application controls appear to be working. If all transactions or supporting data are to be reviewed, substantive testing procedures or tests of financial statement balances should be used. The extent of this testing depends on the audit objectives. For example, an IT auditor usually performs compliance tests over those aspects of an application that cover internal accounting controls related to financial statements. An IT auditor may want to perform compliance tests over other areas, such as the efficiency of administrative controls.

For older applications, tests of inputs and outputs often are quite easy to perform and are not nearly as easy for today's applications, where the auditor often does not encounter a one-to-one relationship between inputs and outputs. Test transaction approaches, discussed next, are often much easier to perform and even more meaningful. Nevertheless, tests of inputs and outputs are sometimes useful for reviews of applications. Compliance test IT audit procedures for an example automated purchasing application are outlined in Exhibit 10.9.

Test Transaction Evaluation Approaches

An IT auditor may want to ascertain that transactions entered into a system are processed correctly. For example, when reviewing a plant floor manufacturing application, an IT

1. Select a series of purchase orders generated by the application reviewed and trace them back to the requirements generated through either the procurement system or authorized manual purchase inputs, determining that all new purchase orders have been properly authorized.
2. From the sample selected in step 1, trace the purchase orders back to established records for vendor terms and prices, resolving any differences.
3. Select and trace a cycle of automated purchase orders to appropriate Web control logs to determine that all documents were transmitted without error and on a timely basis.
4. Using a sample of purchase orders received from log files, determine that vendors are documented through current, signed purchase agreements.
5. Select a sample of receiving reports, and determine that the application is working properly by matching receipts to open purchase orders and accounts payable records.
6. Select a sample of recent accounts payable vouchers and any actual checks generated for parts and materials, tracing the payments to valid receiving reports and purchase orders.
7. Using sample transactions that were either held upon receipt for noncompliance with terms or improper timing, verify that transactions are handled correctly and per established procedures.
8. Balance a full cycle sample of purchase transactions, from the input system providing inputs to the control logs and printed purchase order documents.
9. Investigate any balancing differences and verify the reported errors are valid.
10. Document all exceptions found as potential audit report items.

EXHIBIT 10.9 Automated Purchasing System Compliance Tests Example

auditor might record several shop materials transactions as they are entered on manufacturing floor terminals. After an overnight processing cycle, an IT auditor can verify that those transactions have correctly made adjustments to inventory records and that work-in-process cost reports have been properly updated. This verification can take place by reviewing special retrieval reports against data files. As part of the test transaction process, an IT auditor also can test whether error-screening controls are operating as described. The emphasis here should be on the testing of the error-verification routines within the application. IT audit can select transactions input to an application that appear to be invalid and then trace them through the application to determine that they have been properly reported on exception reports. IT audit also can consider submitting test error transactions to a system to verify that they are being properly rejected by the application.

Other Application Review Techniques

The computer-assisted audit tools and techniques (CAATTs) discussed in Chapter 13 can be useful in reviewing application controls. All too often, IT auditors use these tools to test some accounting control, such as an accounts receivable billing calculation, but do not to evaluate other application controls. Audit software can match files from different periods, identify unusual data items, perform footings and recalculations, or simulate selected functions of an application. Other useful techniques are:

- **Reperformance of application functions or calculations.** This type of test is applicable for both the automated and the manual aspects of application systems. For example, if a fixed-assets application performs automatic depreciation calculations, IT audit can use CAATT processes to recalculate depreciation values for selected transactions as a compliance test.

- **Reviews of program source code.** For applications developed in-house, IT audit can verify that a program performs a certain logic check by verifying the source code. However, this type of compliance test should be used with only the *greatest amount of caution*. Because reading and understanding program source code is complex, it is very easy to miss a program branch around the area being tested. An IT auditor should consider the specialized programs available to compare program source code with the compiled versions in production libraries.
- **Continuous audit monitoring approaches.** IT audit sometimes can arrange to build embedded audit procedures into production applications to allow those applications to flag control or other application exception problems. These techniques also are discussed in Chapter 13 on CAATTs and in Chapter 14 on continuous auditing techniques. This approach goes beyond just auditing an application adding procedures to make it self-auditable on a continuous basis.
- **Observation of procedures.** Observations may be useful when reviewing both automated and manual applications. For example, a remote workstation receiving downloaded data from a central IT system may require extensive manual procedures in order to make the proper download connection. IT audit can observe this on a test basis to determine if the manual procedures are being performed correctly.

Completing the Application Controls Review

Although compliance tests are powerful methods to test application controls, IT audit should be aware that this level of assurance is not absolute. There is a risk that an IT auditor may test an application control and find it to be working when, in fact, it does not normally work as tested. Because of the risks associated with such compliance tests, therefore, IT audit always should be careful to condition any audit reports to management with a comment or warning about the risks of incorrect results due to the limited audit tests. Sometimes the controls tested do not appear to be working correctly because IT audit does not understand some aspects of the application system. IT audit may want to review the application description and identify controls to verify that they are correct. It may be necessary to revise IT audit's understanding of the application controls and then reperform the audit risk assessment procedures.

If IT audit finds that, through compliance testing, the application controls are not working, it probably will be necessary to report these findings. The nature of this report depends very much on the severity of the control weaknesses and the nature of the review. For example, if the application is being reviewed at the request of the audit committee or senior financial management, the identified control weaknesses may prevent them from placing any reliance on the financial results produced by the application. If the control weaknesses are primarily efficiency related or operational, IT audit may want to report them just to IT management for future corrective action.

Applications can be primarily financial or operational. They can be implemented using purchased software, can be custom-developed applications located on in-house systems with extensive database and telecommunications facilities, can operate in a client-server environment, or can exist in numerous other variations. As noted, this diversity makes it difficult to provide one set of audit procedures for all applications. While IT audit can develop a general approach to reviewing most data processing applications, usually it is necessary to tailor that approach to the specific features of a given application. The next section describes how an IT auditor might perform a review of a capital budgeting application operating in a client-server environment with telecommunications links through a network linked to a larger server machine.

APPLICATION REVIEW CASE STUDY: CLIENT-SERVER BUDGETING SYSTEM

As an application review example, assume IT audit has been asked to assess the internal controls over an in-house-developed client-server architecture capital budgeting system. Assume that a financial planning department has developed the capital budgeting analysis portion of this example application using a popular desktop spreadsheet software package. Although the application has been built around a purchased software spreadsheet package, the business users have coded a series of macro instructions for running the programs. The workstation portion of this example system communicates with a server file containing mainframe budgeting system data.

Assume that IT audit has been asked by management to review general controls over both local networks and their client-server computer operations. Following the IT general controls review audit procedures discussed in Chapter 6, IT audit found that general controls in these areas were adequate. That is, users documented their desktop applications; adequate backups of files and programs were performed on server files; password procedures limited access only to authorized personnel; and other good internal control procedures were followed. Among IT audit's recommendations was to place stronger controls over telecommunications access to the local network and to install virus-scanning procedures.

Sometime after that general controls review, this example capital budgeting system was implemented on the enterprise administrative office network. Because this system provides direct input to the corporate budgeting system, management has asked IT audit to review its application controls. After discussing this review request with senior and IT management, IT audit developed these high-level review objectives:

- The spreadsheet capital budgeting system should have good internal accounting controls that are consistent with other enterprise control processes.
- The application should properly make capital budgeting decisions based on both the parameters input to the system and programmed macro formulas.
- The system should provide accurate inputs to the central or corporate budgeting system through the local file server.
- IT security and integrity controls should be secure.
- The capital budgeting system should promote efficiency within the financial planning department.