CHAPTER FIVE
AUDIT SOFTWARE (COMPUTER ASSISTED AUDIT TECHNIQUES)

5.1. Introduction: -

A wide range of software now exists that auditors might find useful during an audit. In this light, one must have an understanding of the nature of the software, the functions it can perform, where it might be used within an audit, and its inherent strengths and limitations. In this chapter, generalized audit software, industry-specific audit software, high-level languages, utility software that auditors often use during the evidence-collection process phase of an audit and specialized software that auditors sometimes must develop and implement to address needs that cannot be met satisfactorily using the said four types audit software is presented.

5.2. Computer Assisted Audit Techniques (CAAT): -

Today’s auditors work constantly with computerized records. It is likely that many audit clients have eliminated or will eliminate a substantial portion of their paper documents and replace them with electronic documents filed only in computerized form. An auditor who is unable to use computerized audit tools effectively will be at a tremendous disadvantage. Therefore, today’s auditor must be equipped with the understanding of alternative tools and approaches to test the operations of computerized systems and gather and analyze data contained in computerized files. The following are the five major reasons for an auditing firm to incorporate computers in their practices: -

1. Increased productivity because auditors can complete routine tasks faster, improve consistency, and focus more on significant issues;

2. Audited tasks that might be impractical or impossible to performed manually can be completed;

3. Reduced cost due to diminished time required to complete the audit work;
4. Competitive advantage gained and client perception of the auditor, the firm, and the quality of the services provided improved;

5. Ability to cope with difficult tasks without corresponding staff increases.

During the course of the audit an IS auditor should obtain sufficient, relevant and useful evidence to effectively achieve the audit objectives. The audit findings and conclusions have to be supported by appropriate analysis and interpretation of this evidence. With systems having different hardware and software environments, different data structure, record formats, processing functions, etc., it is almost impossible for the auditors to collect evidence and analyze the records without a software tool. Owing to resource constraints and the ever changing audit objectives it is almost impossible to quickly develop audit capabilities, without some sort of CAATs.

Pointed underneath the list of selected operational audit tools and techniques:

a. Control Compliance Suite,
   Electronic Discovery & Audit by
   Symantec, Inc., Cupertino, CA

b. SQLsecure by Idera

c. ACL by ACL Services Ltd.,
   Vancouver, BC, Canada

d. IDEA by Audimation Services, Inc.,
   Houston, TX

e. The Number-Audit Sampling by
   Linton Shafer Computer Services,
   Inc., Frederick, MD

f. ADM Plus by Joseph Pleier &
   Associates, Mission Viejo, CA
g. WizWhy™ and WizRule™ by Wizsoft Inc., Syosset, NY


i. RecoveryPAC, RecoveryPAC Web, and RiskPac by CPACS Software Products, Southbury, CT.

j. SAM by Intra Computer, Intra Computer, Inc., Jamica, NY

k. Disaster Recovery System (DRS) by TAMP Computer Systems Inc., Merrick, NY

l. SSA-Name3 by Search America Software, Old Greenwich, CT

m. COD 32, Double Check, and Achieve by IPS of Boston, Braintree, MA


The ICAI Guidance note on CAAT describes CAATs as important tools for the auditor in performing audits. CAATs may be used in performing various auditing procedures including the following:

➢ Tests of details of transactions and balances, for example, the use of audit software for recalculated interest or the extraction of invoices over a certain value from the computer records;
Analytical procedures, for example, identifying inconsistencies or significant fluctuations;

Tests of general controls, for example testing the set-up or configurations of the operating system or access procedures to the program libraries or by using code comparison software to check that the version of the program in use is the version approved by management;

Sampling programs to extract data for audit testing;

Tests of application controls, for example, testing the functionality of a programmed control;

Re-performing calculations performed by the entity’s accounting system.

However, the auditor while selecting the CAAT is faced with certain critical decisions that he may be required to make, while balancing on the quality and cost of audit:

- Use the audit software developed by the client;
- Design and develop his own audit software;
- Use a standard off-the-shelf Generalized Audit Software.

The first two options require the auditor to be technically competent in programming and its methodology, which may not be his area of expertise. Computer audit software, also known as Generalized Audit Programs (GAS) which are readily available and do not require for much expertise from the auditor can be used. The various types of CAATs can be categorized as:

a. Generalized Audit Software;
b. Industry-Specific Audit Software;
c. High Level Languages;
d. Utility Software;
e. Specialized Audit Software.
5.3. **Generalized Audit Software:**

Generalized audit software is the off-the-shelf software that provides a means to gain access to and manipulate data maintained on computer storage media. This software has all the features of mathematical computations, stratification, statistical analysis, sequence check, duplicate check, re-computations, etc. Auditors can obtain evidence directly on the quality of the records produced and maintained by application systems. In turn, their judgments on the quality of the records will enable them to make judgments about the quality of application system that processes these records. GAS cannot perform the audit but can facilitate selection and processing the information as per the clients' requirements. The two most commonly used GAS are ACL (Audit Command Language) and IDEA (Interactive Data Extraction Analysis).

Generalized audit software packages first appeared in the mid-1960s. They were developed by several large public accounting firms to facilitate the audit work they needed to carry out on mainframe computers. Today, however, microcomputer based generalized audit software packages are available. Data is often transferred from mainframe to a microcomputer to enable auditors to work with a generalized audit software package. It has remained the most frequently used computer-assisted auditing tool. Some of the products related to query and analysis are pointed in Table 5.1.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Product</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Access</td>
<td>A database program that provides data selection, analysis, and reporting</td>
</tr>
<tr>
<td>2.</td>
<td>ACL</td>
<td>General audit software that reads files from most formats (e.g. EBCDIC, TXT) and provides data selection, analysis, and reporting</td>
</tr>
<tr>
<td>3.</td>
<td>Excel</td>
<td>Spreadsheet software that provides analysis, calculation, graphing, and reporting</td>
</tr>
<tr>
<td>4.</td>
<td>CA-Examine</td>
<td>A programming language that provides data selection, analysis, and reporting</td>
</tr>
</tbody>
</table>

Table – 5.1
<table>
<thead>
<tr>
<th>S. No.</th>
<th>Product</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.</td>
<td>CA-Easytrieve</td>
<td>A programming language that provides data selection, analysis, and reporting</td>
</tr>
<tr>
<td>6.</td>
<td>Vbasic, C, C++,</td>
<td>A programming language that provides data selection, analysis, and reporting</td>
</tr>
<tr>
<td></td>
<td>JAVA, SQL, Perl</td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>SAS, SPSS</td>
<td>A programming language that provides data selection, analysis, and reporting</td>
</tr>
</tbody>
</table>


5.3.1 **Functional Capabilities of Generalized Audit Software:**

Following are the functions of the generalized audit software:

5.3.1.1 **File Access:**

The file access functions enable files having different data coding schemes, record formats, and file structures to be read. Coding schemes like ASCII, EBCDIC, zoned, packed, and binary often can be read. Records can have fixed or variable formats. Typically, the file structures that can be accessed are sequential, index sequential and random although some packages provide access to more complex structures such as trees and networks. Some generalized audit software packages allow several files to be read simultaneously. Some also provide direct access to files created by several popular database management systems, accounting packages, spreadsheet packages and word processing packages.

5.3.1.2 **File Reorganization:**

The file reorganization functions allow data to be sorted into different orders and data from different files to be merged onto one file. Sorting capabilities are necessary for a variety of purposes. For example, reporting data in a specified order or comparing data on two files. Merging capabilities are needed if data from separate files is to be combined on a separate work file. Function can then be executed on this work file for example, statistical functions or various kinds of calculations.
5.3.1.3 Selection: -

Generalized audit software provides powerful selection capabilities for extracting data that satisfies certain tests. Typically, the Boolean operators AND, OR are provided as well as relational operators EO, GT, LT, NE, GE, LE – that is equal to, greater than, less than, not equal to, greater than or equal to, and less than or equal to. Complex queries containing nested tests can be formulated.

5.3.1.4 Statistical: -

The statistical capabilities of generalized audit software vary from moderately powerful to sophisticated. At a basic level, every n\textsuperscript{th} record can be selected or records can be selected at random. Some packages also provide comprehensive attributes sampling, variables sampling, combined attributes/variables sampling, discovery sampling, stratified sampling and dollar-unit sampling capabilities. Some also provide functions to support analytical review procedures – for example, regression and financial ratio analysis capabilities. Selected data and key analytical review results from prior years can be saved and brought forward to facilitate analytical review in subsequent years. Generalized audit software also can be designed to provide input to separate statistical and financial modeling software where more powerful capabilities are required.

5.3.1.5 Arithmetic: -

Generalized audit software provides the full set of arithmetic operators enabling work fields to be computed, the arithmetic accuracy of data to be checked, control totals to be produced, and so on. For example, net pay calculations for a payroll file can be recomputed, or files can be cross-footed. Often the calculations can be made based on data from more than one input record. Calculated fields can be stored and then used in subsequent calculations.

5.3.1.6 Stratification or Frequency Analysis: -

Generalized audit software packages often provide good capabilities with respect to stratification and frequency analysis. Different types of stratification and frequency
analysis can be undertaken. For example, the frequency of accounts receivable balances in certain classes can be determined: Rs. 0 – Rs. 200, Rs. 201 – Rs. 400, Rs. 401 – Rs. 600 and so on.

5.3.1.7 File Creation and Updating: -
Some generalized audit software packages allow work files to be created and updated. For example, the output could contain samples of input file records or user-defined records that include fields extracted from input records or calculated fields. In some cases, the output files can be written in formats that are suitable for input into widely used database management software, spreadsheet software, or word processing software.

5.3.1.8 Reporting: -
Comprehensive reporting facilities are often available in generalized audit software packages. For example, free-form reports can be produced that allow auditors to control the title of the report, content of column headings, width of columns in the report, levels of subtotals, number of detail lines, page footers and page headers, and formatting of fields. Some reports that contain data that most auditors will require during an audit are produced automatically – for example, reports containing control totals, record counts, negative amounts, and blank or zero fields.

5.3.2 Audit Tasks that can be Accomplished Using Generalized Audit Software: -
Auditors can combine the functional capabilities of generalized audit software to accomplish several audit tasks: -

1. Examine the quality of data,
2. Examine the quality of system processes,
3. Examine the existence of the entities, and
4. Undertake analytical review.
5.3.2.1 **Examine the Quality of Data:**

Auditors can use the functional capabilities of generalized audit software to examine the existence, accuracy, completeness, consistency, and timeliness of data maintained on computer storage media. Consider the following examples:

a. Records for various fixed assets can be retrieved to see if, in fact, the records exist.

b. The calculation of sales discounts can be checked for accuracy.

c. The address field for customers in an accounts receivable file can be examined to see if it contains blanks.

d. Records on the personnel file and the payroll file can be compared for consistency.

e. A file of share can be checked to determine the last time it was updated.

Auditors might examine the quality of data maintained in application system files or in an organization’s database for two reasons. First, the quality of the data reflects the quality of the application system that processes the data. For example, if the address field in a debtor’s record is blank, we should question the adequacy of the validation processes contained in the system. Second, the quality of data reflects the quality of the personnel who developed and maintain the application system and the quality of the personnel who use the system.

5.3.2.2 **Examine the Quality of System Processes:**

Even though the quality of data in an application system might be high, the quality of system processes still could be low from the view point of achieving the objectives of an organization. For example, the data in accounts receivable file might be accurate, complete, consistent, and timely. A substantial number of overdue accounts might exist, however, which reflects adversely on both the accounts receivable application
system and the personnel who use the system. The system might not be producing adequate management reports to enable timely collection of receivables. Alternatively, the system might be producing adequate reports. Nonetheless, accounts receivable personnel may not be using the reports or not properly following up on collections.

Besides examining the quality of data, auditors can use generalized audit software to examine the quality of processes in other ways. For example, in the accounts receivable example, auditors could use it to age the accounts receivable file to determine whether debtors were paying their accounts on a timely basis. Similarly, auditors could use generalized audit software to calculate inventory turnover statistics as a basis for identifying obsolete inventory. If they identified substantial amounts of obsolete inventory, auditors should then question the adequacy of system processes for managing inventory. Moreover, auditors should also question whether inventory is overvalued.

Still another way of using generalized audit software to examine the quality of the processes in an application system is via a technique called parallel simulation. It involves auditors writing a program to replicate those application processes that are critical to an audit opinion and using this program to reprocess application system data. The results produced via the simulation program are then compared with the results produced by the application system. Any discrepancies identified from the basis for follow-up work by the auditor.

5.3.2.3 Examine the Existence of the Entities: -

Data could exist and be accurate, complete, and consistent. It might not represent an object in the real world; however, for example, it might represent a bogus insurance policy or an inventory item that no longer exists. Auditors must determine, therefore, whether the entities that the data purports to describe really exist. The statistical sampling capabilities of generalized audit software provide an important means of doing this. For example, auditors can use these capabilities to select a sample of debtors for confirmation or a sample of inventory for physical observation. The
powerful reporting capabilities of generalized audit software can then be used to print confirmations in the form required for mailing to debtors or to sort and print inventory data in a way that will facilitate auditors’ physical counts of inventory. They can then input the results obtained from their confirmations or physical inventory work to generalized audit software to obtain probabilistic statements about the number of errors that is likely to exist in the accounts.

5.3.2.4 Undertake Analytical Review:

Analytical review is the process of obtaining key ratios and totals from an organization’s data for comparison with previous years’ ratios and totals or industry-wide ratios and totals. The information obtained from analytical review is used to support or question preliminary audit conclusions based on system reviews and other substantive tests. For example, a decline in the working capital ratio of an organization might be used to support a preliminary audit conclusion that the ongoing viability of the organization is at risk.

Auditors can use generalized audit software to support analytical review work in several ways. First, they can use generalized audit software to extract data required for analytical from an organization’s database or an outside database and to prepare various ratios and totals; second, if generalized audit software provides regression analysis capabilities, auditors can use the software to examine firm and industry trends. Alternatively, they can use generalized audit software to prepare data in format suitable for input to another package that provides regression capabilities or other kinds of modeling capabilities required. Third, auditors can use generalized audit software to maintain a database of key data and key indicators across time.

5.3.3 Functional Limitations of Generalized Audit Software:

To generalized audit software effectively and efficiently, auditors must understand both its capabilities and its limitations. Following are the limitations:
a. Generalized audit software permits auditors to undertake only ex post auditing and not concurrent auditing.

b. Generalized audit software has only limited capabilities for verifying processing logic.

c. It is difficult for auditors to determine the application system’s propensity for error using generalized audit software.

5.3.3.1 Ex Post Auditing Only: -

Generalized audit software enables evidence to be collected only on the state of an application system after the fact. In other words, the software examines the quality of data after it has been processed. Even if auditors use generalized audit software to undertake parallel simulation, the results produced by the parallel simulation program are checked against a set of existing results produced by the application system. Thus, some time lag will occur between an application system error occurring and its possible identification using generalized audit software. In some cases this elapsed time could be substantial if the application system is not audited on a regular basis.

For some types of systems, timely identification of errors could be critical. For example, consider a situation in which multiple online users access a shared database and an error in such a situation is called concurrent error. Unless an error that occurs in a data item is discovered quickly, it could permeate the database and cause several incorrect decisions to be made by users. Timely identification and correction of the error is therefore critical. Auditors must use specialized rather than generalized audit software, however, to implement concurrent auditing techniques.

5.3.3.2 Limited Ability to Verify Processing Logic: -

Often the tests performed with generalized audit software involve “live” data - that is, data captured and processed by application system during the normal course of business. The limitations of using data to test application systems, however, are well known. The data might not manifest the exceptional conditions that occur occasionally within the application. As a result, the application system’s capability to
handle these exceptional conditions accurately and completely is not tested. To overcome this problem, the test data must be designed specifically to determine how the application system handles exceptional conditions.

5.3.3 Limited Ability to Determine Propensity for Error:  
Systems can be designed and implemented in ways that allow them, at least to some extent, to accommodate change. For example, database management systems can be used to isolate certain types of changes to the database design from the application systems that access the database. Alternatively, application systems can be designed and implemented in ways that cause them to degenerate quickly when change occurs.

Auditors must be concerned with whether application systems have been designed appropriately to accommodate change. Otherwise, there are higher inherent risks associated with application systems because errors are more likely to result when changes to the systems must be undertaken. Unfortunately, auditors can obtain little evidence using generalized audit software on an application system’s capability to accommodate change. Instead, the evidence must be obtained in other ways – for example, reviewing the management control framework, examining the ways application systems are designed, and examining the ways program code is written.

5.4. Industry-Specific Audit Software:  
Some types of audit software packages are now available that are oriented toward specific industries – for example, the financial services, health care, and insurance industries. The packages are still generalized because they provide auditors with high-level languages that can be used to invoke a wide range of functions. They differ from the types of audit software examined previously, however, in two ways. First, because they are oriented toward a particular industry, they provide high-level commands that invoke common audit functions needed within the industry. For example, in the banking industry, they might use a single command to invoke logic that would check for account kiting. If generalized audit software were to be used to check for kiting, several commands might be required to express the logic needed to execute the
various tests. Second, industry-specific audit software could have been developed to access data maintained by a specific generalized application package that is used extensively within the industry. Accordingly, the file, record, and field definitions used by the application package could be built into the audit software package; that is, auditors do not have to provide these definitions each time they want to run the package.

The CAPS package developed by Brisbane-based Kendalls Chartered Accountants is an example of an industry-specific audit software package. It has been designed for auditors of financial institutions. As such, it provides high-level commands to invoke functions that they will need. In addition, CAPS has been written to access the data maintained by two widely used generalized application packages within the finance industry. Indeed, CAPS cannot be used unless auditors employ one of these two packages for their basic application processing. If the auditee uses one of those packages, however, CAPS provides nine major sets of audit capabilities:

a. **Loan Arrears Audit:**
CAPS can be used to evaluate the movements in loan arrears on a member's loan balance throughout a specified period. For example, a report is provided showing any case in which a new disbursement has occurred in spite of the loan being in arrears. Using this type of information, auditors could assess the auditee's controls over loan arrears.

b. **Interest Audit:**
This module recalculates all interest on member loans and savings accounts to provide an independent check on calculations carried out by the application system.

c. **Term Deposit Interest Audit:**
This module recalculates total term deposit interest to a specified date to provide an independent check on calculations carried out by the application system.
d. **Member Ledger Balances Audit:**

This module provides several functions that assist auditors to evaluate the veracity of member ledger balances. For example, it provides summarized information on each loans, savings, and investments ledger; it allows stratified sampling of member ledger balances for confirmation purposes; and it provides routines to statistically evaluate the results of a confirmation of members.

e. **Member Ledger Transactions Audit:**

This module examines ledger transactions for evidences of unusual circumstances. For example, it identifies transaction, values outside a specified range; it identifies when a disproportionate number of a particular transaction type has occurred; and it selects transactions randomly for audit scrutiny.

f. **Member Biographical Audit:**

This module examines the reasonableness of various demographic and personal data held about a member. For example, it looks for member names without vowels (unusual names); it tests for post codes (pin (postal index number) codes / zip (zone improvement plan) codes) outside a particular range; and it tests for members who have a post office box number as a primary address.

g. **Dormancy Audit:**

This module identifies member accounts that are dormant and that, as a consequence, bear a greater risk of fraudulent or unauthorized transactions remaining undetected for some time. The module retains a separate file of dormant accounts and provides a report on changes to the file when subsequent dormancy audits are conducted.

h. **Incompatible Duties Audit:**

This module allows the set of transactions that different operators / tellers are allowed to execute to be defined. It will then check transaction log files to determine whether any operators are executing transactions that manifest inadequate separation of duties.
i. **Legislative Compliance Audit:**

This module determines whether the financial position of an organization complies with legislative requirements. For example, a credit union might have to ensure that the proportions of its loans maturing within 3 months, 6 months, 9 months, 12 months and greater than 12 months fall within certain ranges. Otherwise, it will be in breach of legislation.

The primary advantages of industry-specific audit software over generalized audit software are that it runs more efficiently and that it is easier to use because it incorporates higher-level functions. The primary disadvantage is that, it has a more limited domain of application than generalized audit software. As such, it tends to be more useful for internal auditors or external auditors who perform a large number of audits within a specific industry.

5.5. **High-Level Languages:**

Besides generalized audit software, auditors can often use a high-level language to gain access to data and manipulate this data. In particular, many auditors use fourth-generation programming languages, such as SQL and QBE, and generalized statistical software, such as SPSS and SAS, to collect evidence on system reliability. Some of the fourth-generation languages and other support tools are:

- MS Office;
- Perl;
- SAP;
- QMF;
- ACL;
- IDEA;
- Oracle;
- Webmetrics 3.0;
- C++;
- V-Basic;
- DMS;
- Asset;
- JAVA, etc.
Fourth generation languages have proved useful to auditors' work for several reasons. First, most functions incorporated within generalized audit software packages are also included within fourth-generation languages. For example, auditors can use fourth-generation languages to select data from files that satisfy certain criteria and to format this data for reporting purposes. They might have weaker capabilities in certain few areas – for example, statistical sampling capabilities. Often auditors can overcome these difficulties, however, by using “macros”, which allow them to write programs to perform particular functions and then to invoke these programs with a single command.

Second, for the types of functions auditors might want to perform, fourth generation languages could be more user friendly than generalized audit software. For example, a fourth-generation language might provide then with more flexible reporting capabilities. Auditors might also be able to avoid difficult downloading of data from one computer to another computer or troublesome conversion of one file or data format to another file or data format.

Third, if auditors use a fourth-generation language that is employed extensively throughout the organization audited, they are likely to able to get good support to overcome any difficulties they might encounter. For example, if the organization uses relational database and SQL, many persons within the organization should be able to assist auditors if they have problems using SQL to access and manipulate data in the database.

Many auditors have also become more frequent users of statistical packages because they now place increased reliance on analytical review as a diagnostic tool in the conduct of audits. In some generalized audit software packages, the statistical capabilities provided are fairly basic. They are oriented primarily toward support of statistical sampling activities. Analytical review often relies on using other statistical models, however, some of which are complex and require substantial computational support.
5.6. **Utility Software:**

Utility software is software that performs fairly specific functions that are needed frequently, often by a large number of users, during the operation of computer systems. For example, they include copy programs, sort programs, disk search programs, and disk formatting programs. They often come as a part of the suite of programs provided with major system software, such as operating systems, database management systems, fourth-generation languages, or data communication software. Much independent utility software has now been developed, however, it can be purchased to undertake functions that cannot be accomplished using the utility programs provided with system software or alternatively to undertake functions more effectively and efficiently than the utility programs provided with system software. Some also exists as freeware and might be downloaded, from a site on the Internet.

Auditors use utility software for five reasons:

First, utility software might have been developed to perform a specific security-or-integrity-related function. For example, auditors might use a utility program to check for viruses on a disk.

Second, before auditors can use generalized audit software or other types of audit software, they might need to format and download data using utility software.

Third, utility software might perform functions that cannot be performed using generalized audit software or other audit software available. For example, auditors might use a utility program to try to recover a damaged disk file that contains data that is material to the audit. It is unlikely that audit software will be able to perform this function.

Fourth, utility software might accomplish audit tasks more effectively and more efficiently than audit software. For example, it might be possible to select certain
kinds of data and print a report using generalized audit software. Utility software might perform the same function but consume fewer resources and prepare better-formatted reports.

Fifth, auditors might use utility software to assist with the development of new audit software. For example, they might seek to develop audit modules that they can embed in application systems to collect evidence at the same time that application system processing occurs. Auditors might use utility software to help test whether the modules work accurately and completely before they release the modules into production.

Because many utility software packages are now available, auditors might often have difficulty identifying what software exists, where it is located, how it can be procured, and what functions it performs. Some operating system vendors have produced documentation that describes utility software that they and other organizations have developed which can assist auditors in their work. Various Usenet groups on the Internet will also provide information about utility software that might be useful. It will help, also, if auditors understand the major types of utility software that exist so they can perhaps pinpoint better the software they need.

5.7. **Specialized Audit Software:**

Specialized audit software is software written in a procedure or problem oriented language to fulfill a specific set of audit tasks. The term “specialized” does not mean the software performs only a narrow range of functions. Indeed, in some cases the software has extensive functionality. Rather, specialized means auditors have developed and implemented the software where the purposes and users of the software are well defined before the software is written. On the other hand, with generalized software, the specific tasks to be undertaken by the software and the identity of users will not be known at the outset.
5.7.1 Reasons for Developing Specialized Audit Software:

There are six reasons auditors might develop specialized audit software:

5.7.1.1 Unavailability of Alternative Software:

Occasionally, auditors might encounter situations in which no generalized software is available to perform the audit procedures. For example, the auditee might have developed or purchased some type of specialized hardware platform on which only a minimal suite of software will run.

5.7.1.2 Functional Limitations of Alternative Software:

Even if auditors have generalized software available to perform an audit task, the functionality might be limited. For example, government auditors sometimes undertake complex information processing activities to check for errors and irregularities. They match data from tax returns, bank accounts, share transactions, welfare payments, and so on, to identify whether citizens are defrauding their government. The generalized software available to government auditors might not be capable of processing the large number of files that must be matched concurrently nor handling the complex data formats and file structures that have been used.

5.7.1.3 Efficiency Considerations:

In some cases, the audit tasks to be undertaken consume substantial resources, perhaps because auditors have to access large databases or have to perform audit tasks frequently. For example, in the complex matching task sometimes undertaken by government auditors examined previously, processing efficiency is often a primary objective. The matching task can be very costly because large, complex data files have to be processed. In this light, government auditors often develop specialized audit software because it will perform the matching task more efficiently than generalized software.
5.7.1.4 Increased Understanding of Systems:

Sometimes the systems to be audited are complex. Nonetheless, it is important that auditors gain a proper understanding of the system as a basis for conducting the audit. One way that they might seek to gain this understanding is to prepare program specifications and to write the source code for specialized audit software. In the case of the computer-matching example examined previously, government auditors might gain valuable insights into the application systems that process the files used in the matching task if they participate in the development and implementation of the specialized software used to carry out the matching task.

5.7.1.5 Opportunity for Easy Implementation:

Opportunities sometimes exist to develop and implement specialized audit software quickly and easily. For example, auditors might be able to insert a few instructions in an application system that gathers data that is critical to a judgment on the reliability of controls in an application system.

5.7.1.6 Increased Auditor Independence / Respect:

To the extent auditors develop their own software and are not reliant on the auditee to provide software or staff support, they are more independent in the conduct of their audit. Moreover, auditors have an opportunity to demonstrate professional competence to the auditee. As a result, the auditee might have increased respect for their work.

One important area where auditors often have to prepare specialized audit software is in the development and implementation of concurrent auditing techniques. Concurrent auditing techniques collect audit evidence at the same time as the application system is processing production of data. They require audit hooks, modules, or routines to be embedded in the application system to select the evidence required. These are often implemented via specialized program code.
5.7.2 Development and Implementation of Specialized Audit Software: -
Specialized audit software can be developed and implemented in three ways. First, auditors can take total responsibility for developing and implementing the software themselves. This approach allows auditors to exercise a high level of control over the software. To produce high-quality software, however, auditors must possess good analysis, design, and programming skills. Second, internal auditors can ask programmers in their own organization to develop and implement the software. Alternatively, external auditors can ask programmers in the client organization to develop and implement the software. Third, auditors could ask an outside software vendor to prepare the software. Auditors might adopt this approach if the software is quite sensitive. Though the costs might be higher, using the services of an independent third party provides extra assurance that integrity violations have not occurred during the development and implementation process.

Whatever the approach auditors use to develop and implement specialized audit software, they must exercise careful control over the development and implementation process to ensure that the software meets their objectives and the integrity of the software is preserved. Auditors can exercise most control when they prepare the software themselves. If auditors use other personnel to prepare software, however, they should still take responsibility for preparing program specifications, managing the programming process, performing acceptance testing, and preparing user documentation. Unless auditors perform these tasks, they must be circumspect about placing reliance on the integrity of the program.

Over the years, number of software has been developed to assist auditors in their audit work. The following examples illustrate the varied nature of the software: -

a. Burns and Loebbecke (1975) have shown how software can be written to simulate the operations of an internal control system. This software can be used to estimate the dollar error that might occur in accounts as a result of weaknesses in the internal control system.
b. Gamer and Tsui (1985) have developed questionnaire generator software. Users indicate potential weaknesses in an internal control system. The software then suggests questions and issues that might be pursued to tease out the nature and extent of the internal control weaknesses.

c. Bailey et al. (1985) have developed a system that allows users to describe the characteristics of an internal control system. The system produces, as output, a model of the internal control system. It also allows users to ask questions about various characteristics of the internal control system to make an assessment about whether various internal control objectives are being met.

d. Hansen and Messier (1984) have proposed the use of a relational database management system to represent the complex interrelationships among controls, their locations, vulnerabilities that result if they are absent, and possible exposures. This data can be incorporated into a decision support system that will assist users to make decisions about where they should collect evidence on the reliability of controls.

References:

6. www.acl.com
7. www.cit.icai.org
8. www.icai.org
9. www.idea.com
10. www.isaca.org