3.1 Introduction: -

Systems development has responsibility for those functions concerned with analyzing, designing, building, implementing and maintaining information systems. In many ways, how we undertake these functions is still an art. Although we have made substantial progress in terms of providing theory and heuristics to guide our practice, good systems development work still relies on the insights, intuition and experience of individual systems analysts and designers.

This chapter describes how we can undertake an audit of the systems development management subsystem. We begin by considering three different ways to approach the audit: as a participant in the systems development process, a post implementation reviewer of a specific application system or a reviewer of the systems development process in general. Further we examine the phases of system development life-cycle approach in addition to role that may be performed through the auditor in SDLC phases. Furthermore, the major tasks performed during systems development, the controls exercised over these tasks and the ways we might evaluate the reliability of these controls. And finally, the information system auditor’s role in system development is depicted.

3.2 Approaches to Auditing Systems Development: -

Historically, a major debate among information systems auditors has focused on the question of whether they should become involved as a member of the project team during the systems development process. Those who favour involvement point out that errors are more costly to correct at later stages in the systems development process. Thus, auditors can play a valuable role by indicating where systems development deficiencies exist at an early stage. On the other hand, those who oppose involvement argue their independence will be undermined if they must later evaluate systems they have helped design and implement. They contend that information
systems auditors should evaluate the systems development process only in an ex post review capacity.

Auditors might conduct three types of audits of the systems development process:

Concurrent audit: Auditors members of the systems development team. They assist the team in improving the quality of systems development for the specific system they are building and implementing.

Post implementation audit: Auditors seek to help an organization learn from its experiences in the development of a specific application system. In addition, they might be evaluating whether the system need to be scrapped, continued or modified in some way.

General audit: Auditors evaluate system development controls overall. They seek to determine whether they can reduce the extent of substantive testing needed to form an audit opinion about management's assertions relating to the financial statements or systems effectiveness and efficiency.

An external auditor is more likely to undertake general audits rather than concurrent or post implementation audits of the systems development process. Nevertheless, clients might ask external auditors to participate in the systems development process if they believe their involvement will reduce overall audit costs or they value their advice. Similarly, they may ask external auditors to undertake a post implementation review if they believe they can provide cost-effective advice. For internal auditors, management might require that they participate in the development of material application systems or undertake post implementation reviews of material application systems as a matter of course.

Both external and internal auditors must strive to preserve their independence. They should not conduct ex post reviews of any systems in which they were a member of
the systems development team. They cannot independently evaluate their own work! They can also protect their independence by ensuring that they have sufficient knowledge and skills to be able to form independent judgments about the quality of systems development work. With careful planning, auditors can mitigate some of the concerns that arise about participation in the systems development process.

3.3 **Systems Development Life-Cycle Approach:**

Traditionally, systems development personnel have thought about the systems development process in terms of a life cycle comprising various major phases. The life-cycle approach arose from early efforts to apply project management techniques to the systems development process. Historically, many systems were characterized by massive cost overruns, inadequate economic evaluations, inadequate system design, management abdication, poor communications, inadequate direction and so on. The models used in system development are waterfall model, socio-technical design approach, soft system approach, prototype approach, rapid application approach and contingency approach etc.

The traditional SDLC has the following seven sequential phases. Each of the earlier phase provides a basis for the subsequent phase of SDLC. The underneath table points out the traditional SDLC phases.

**TABLE 3.1**

<table>
<thead>
<tr>
<th>Phase No.</th>
<th>Phase Name</th>
<th>Nature of Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Feasibility Study</td>
<td>Determining and evaluating the strategic benefits of the system and ensure that the solution fits the business strategy. Includes cost-benefit analysis of the proposed system.</td>
</tr>
<tr>
<td>2.</td>
<td>System Requirement Analysis</td>
<td>Analyzing the type of the system on the basis of the users' requirements.</td>
</tr>
<tr>
<td>3.</td>
<td>Design</td>
<td>Designing the system in terms of user interface, data storage and data processing functions on the basis of the requirement phase by developing the system flow charts, system and data flow diagrams, screens and reports etc.</td>
</tr>
<tr>
<td>4.</td>
<td>Development/Programming</td>
<td>Programming the system as designed and conduct the continuous testing and debugging.</td>
</tr>
<tr>
<td>Phase No.</td>
<td>Phase Name</td>
<td>Nature of Activity</td>
</tr>
<tr>
<td>----------</td>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5.</td>
<td>Testing</td>
<td>Various kind of testing is conducted before the developed system is implemented. The testing include Unit testing, Integration testing, and System testing etc.</td>
</tr>
<tr>
<td>6.</td>
<td>Implementation</td>
<td>Final testing and quality and controls audit, acceptance by management and user before migration of the system to live environment and data conversion from legacy system to the new system.</td>
</tr>
<tr>
<td>7.</td>
<td>Post Implementation</td>
<td>Continuous evaluation of the system as it functions in the live environment and its updation.</td>
</tr>
</tbody>
</table>

For any kind of a partnership involving auditors, users, and IS management, it is important that the organization establish a formal procedure for the development of a system. Without a formal SDLC, complete with defined phases of development and specified points for review and evaluation, the auditor’s job is much more difficult and recommendations are not as readily accepted.

- Auditor influence is significantly increased when there are formal procedures and required guidelines identifying each phase and project deliverable in the SDLC and the extent of auditor involvement.
- Auditors will be able to review all relevant areas and phases of the SDLC, identify any missing area for the development team, and report independently to management on the adherence to planned objectives and procedures.
- Auditors can identify selected parts of the system and become involved in the technical aspects based on their skills and abilities.
- Auditors can provide an evaluation of the methods and techniques applied in the systems development process.

3.3.1 **Analysis:**

The project team should expend considerable effort toward the analysis of the business problem and what the system is to produce without initially attempting to develop the design of the system. The auditor should observe that the primary responsibility is not to develop a product but to satisfy the user. Often, the user does not understand what is truly needed. Only by understanding the user’s business, its problems, goals, constraints, weaknesses, and strengths can the project team deliver the product the user needs.
3.3.2 **Design:**

The auditor may review the design work to make sure that the user's requirements are met. The system's design may also be reviewed for any possible exposures or forgotten controls and for adherence to company standards. If an exposure is found, the auditor should recommend the appropriate controls or procedures.

A technique that brings users project team members together for an intensive workshop in which they create a system proposal into a detail design is called Joint Application Design (JAD). Usually a trained JAD facilitator, having some claim to neutrality, takes the group through formatted discussions of the system. The auditor may be an active participant in this process. The result of the JAD session is a user view of the system for further development. This is an excellent setting for the discussion of the advantages and cost effectiveness of controls. In addition, analysis time is compressed, discrepancies resolved, specification errors reduced, and communication greatly enhanced. Auditor's can review deliverables and recommend application controls.

3.3.3 **Construction:**

The auditor may review the new system's programs to verify compliance with programming standards. These standards help to ensure that all the code has a similar structure, tracking dependencies and making maintenance easier. The auditor may review a sample of programs to verify that the standards are being followed and that the programs conform to system design. In addition, programs may be checked for possible control exposures and for the placement of proper controls per design. If it is determined that controls are needed, the auditor should make recommendations, following the same criteria that were used during the design phase. During this phase, however, cost and time factors must be carefully considered because the cost of the changing programs to include controls increases as the project progresses.

3.3.4 **Testing:**

The auditor may be called on to assure management that both developers and users have thoroughly tested to ensure that it:

58
> Possesses the built-in controls necessary to provide reasonable assurance of proper operation

> Provides the capability to track events through the systems and thus supports audit review of the system in operation

> Meets the needs of the user and management

If the level of testing does not meet standards, the auditor must notify the development team or management who will then take corrective action.

3.3.5 Documentation: -

The auditor may review system, user, or operating documentation for completeness and accuracy. The maintenance programmers and users should easily understand the documentation. For instance, diagrams of information flow and samples of possible input documents/screens and output reports enhance understanding of the system.

3.3.6 Implementation: -

The auditor may review the implementation strategy, communication and training material, documentation, conversion procedures, and production readiness. Production readiness may include evaluating the readiness of the system in relation to the results of testing, the readiness of production support programmers, computer operations, and users in terms of training, and the readiness of the help desk with trained staff and a problem-tracking process.

3.3.7 Postimplementation: -

Once the system is in production, the auditor may survey users to evaluate the effectiveness of application from a workflow perspective, review error detection and correction procedures to confirm they are working as intended, or perform tests of data to confirm completeness of transaction processing and audit trial.
3.3.8 Change Control: -
Changes to a system in production come from two sources: problems not detected during testing and change in user requirements. A change control review would evaluate problems are reported, tracked, prioritized, and resolved, and whether changes are authorized, tested, documented, and communicated.

3.3.9 Application Controls: -
When reviewing the system development phases, the auditor examines application and manual control points. The user department is responsible for specifying the needed controls and the systems analysts and programmers are responsible for implementing these controls. Although the responsibility for the auditability and controls in the new system lies with the user departments, system analyst and programmers, these groups may not have the expertise to design adequate controls into the new system. The auditor should interact with each of these groups during the development process to ensure the adequacy of audit and control provisions. The completion of each of the phases usually corresponds to the key points for auditor involvement.

3.3.10 Communication: -
The first area to communicate is the auditor' role in the systems development project. It is very important to make sure that the management and development teams' expectations of the auditor's role are understood and communicated to all participants. To influence the systems development effort, the auditor must develop an open line of communication with both the management and users. In addition, the auditor must develop a good working relationship with analysts and programmers. Although the auditor should cultivate good working relationships with all the groups that have design responsibilities, the auditor must remain independent.

3.3.11 Recommendations: -
Throughout the development project, the auditor will be making control recommendations. Depending on the organization's culture, these recommendations
may need to be handled informally by reviewing designs with the project team or formally by presenting recommendations to the steering committee. In either case, the auditor must always consider the value of the control recommendation versus cost of implementing the control. Also, recommendations should be specific, identifying the problem and not the symptom. This allows the proper controls to be implemented and tested.

3.3.12 Audit Report:

Depending on the audit scope and the length of the project, interim reports may be needed at the completion of major phases in the development process. Key reporting points include:

- Planning: Adequacy of the project plan and cost/benefit analysis.
- Design: Adequacy of the controls and auditability of the design.
- Testing: Adequacy of the test strategy and completeness of the test objectives.
- Implementation: Readiness of the system and user components for production.
- Postimplementation: Effectiveness and efficiency of the live system and whether the initial system criteria were met.

If the auditor becomes part of the development team, the team manager would become the auditor's management, and the auditor's reports and recommendations might be modified before sent to higher management. To remain objective, the auditor should issue reports independent of the project team.

3.4 Evaluating the Major Phases in the Systems Development Process:

In the following subsections we examine the tasks that must be undertaken and the controls that may be important in 13 major systems development phases pointed underneath.

1. Problem/opportunity definition;
2. Management of the change Process;
3. Entry and feasibility assessment;
3.4.1 **Problem/Opportunity Definition:**

Problems and opportunities that might be amenable to information systems support can be recognized in two ways. First, they can be conceived through formal processes associated with the preparation of an information systems plan. Second, they can be conceived fortuitously. During the problem/opportunity definition phase, the stakeholders must attempt to come to some understanding of the nature of the problem or opportunity they are addressing. Is the problem or opportunity well or ill structured? Does it have implications for a small or a large number of people? Will possible solutions have a large impact on the organizational structure and jobs? Will new technology most likely be needed to support possible solutions? Auditors should have the following types of concerns about the activities carried out in this phase:

1. If possible information systems solutions to the problem or opportunity will be material in terms of size or impact, have formal terms of reference been prepared? If so, have they been approved by a steering committee or well constituted project committee?

2. If possible information systems solutions will have a major impact on task systems or social systems, what level of acceptance exists among the
stakeholders on the need for change? Do the terms of reference consider the need for consultation and negotiation?

3. If there is a high level of requirement uncertainty or technological uncertainty surrounding possible, solutions to the problem or opportunity do the terms of reference take into account strategies that might help alleviate the uncertainty?

4. Do the stakeholders agree on the definition of the problem or opportunity? If they disagreed at the outset, what approaches were used to try to reach consensus?

During the problem/opportunity definition stage, therefore, auditors are concerned to see that the stakeholders have reached agreement on the problem or opportunity and that they have an understanding of the threats to asset safeguarding, data integrity, system effectiveness and system efficiency associated with possible solutions. Their concerns will be minor if the problem or opportunity and possible solutions are local and straightforward. They increase, however, as the problem or opportunity becomes more ill structured and possible solutions have a wider impact.

3.4.2 Management of the Change Process: -

Management of the change process runs parallel to all other phases. The change process starts at the initial conception of the system and continues until the new system is running and the organization has adjusted to the new system. Management of the change process involves two major tasks: project management and change facilitation. Project management involves addressing such matters as budgeting, exception reporting, checkpoints, and user signoffs. The way in which project management is undertaken should vary, however, depending on the type of system to be developed and implemented. For example, if the project is uncertain, the external coordination mechanisms must be established to increase the likelihood requirements to be specified correctly. More users might be placed on the design team, for example, to improve communications among the stakeholders. If, on the other hand, the project
is straightforward, traditional internal coordination mechanisms like PERT charts may be used.

Management of the change-facilitating aspects of systems development becomes more critical as possible solutions are likely to have a greater impact on organizational structures and jobs. Current prescriptive models for facilitating change in the systems development process tend to be some adaptation of the models of organizational change. Three major classes of activities are required:

<table>
<thead>
<tr>
<th>Class of Activities</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unfreezing the organization</td>
<td>Preparing the organization for change; providing feedback to the organizational members on their attitudes and behaviors; using techniques such as education, participatory decision making and command to promote the need for change. Unfreezing activities help to avoid having to impose change on stakeholders.</td>
</tr>
<tr>
<td>Moving the organization</td>
<td>Chang over to new system</td>
</tr>
<tr>
<td>Refreezing the organization</td>
<td>Helping system users adapt to their new roles by providing positive feedback on their changed attitudes and behaviors. Refreezing activities make it more difficult for stakeholders to revert to their old attitudes and behavioral patterns.</td>
</tr>
</tbody>
</table>

During this phase, auditors must evaluate the quality of decisions made about project management and change facilitation. Again, they must adopt a contingency perspective. If the proposed system is small, has a localized impact, and will be developed by end users employing a high-level language, project management and
change facilitation concerns are unlikely to be material. On the other hand, if the proposed system is large, high levels of requirements and technological uncertainty exists and organizational structures and jobs will be affected significantly, the decisions made about project management and change facilitation will be a material concern.

3.4.3 **Entry and Feasibility Assessment:**

The purpose of the entry and feasibility assessment phase is to obtain a commitment to change and to evaluate whether cost effective solutions are available to address the problem or opportunity that has been identified. If entry is successful, the designer can then carry out a preliminary study to evaluate the feasibility of the new system using four criteria:

- **Technical feasibility:** Is the available technology sufficient to support the proposed project? Can the technology be acquired or developed?

- **Operational feasibility:** Can the input data be collected for the system? Is the output usable?

- **Economic feasibility:** Do the benefits of the system exceed the costs?

- **Behavioral feasibility:** What impact will the system have on the users' quality of working life?

The specific techniques used to evaluate the feasibility of systems should vary depending on the type of system being proposed. For example, if the proposed system is relatively straightforward and the likely benefits and costs are clear, net present value analysis might be used to assess economic feasibility. On the other hand, if substantial uncertainty surrounds the system, an approach such as value analysis might be used to assess feasibility. When value analysis is used, seed funding is initially allocated to a system to build a prototype. When users gain experience with the prototype, better estimates of costs and benefits can be obtained. If the anticipated benefits of proceeding further appear to exceed the costs, additional funding is allocated to extend the prototype. In light of experimentation with the extended
prototype, additional funding might be provided. This process proceeds iteratively until the benefits no longer appear to exceed the costs.

3.4.4 Analysis of the Existing System: -

When a new system is proposed, in some cases it will replace an existing system. Designers might need to understand the existing system if they are to undertake high-quality work in developing and implementing a new system. In addition, the strategic requirements for the new system could be based on the existing system's strengths and weaknesses. Analysis of the existing system usually involves two major tasks: (1) studying the existing organizational history, structure and culture and (2) studying the existing product and information flows.

3.4.4.1 Studying the Existing Organizational History, Structure and Culture: -

Designers might need to study the existing organizational history, structure and culture to gain an understanding of the social and task systems in place, the ways these systems are coupled and the willingness of stakeholders to change. Auditors should be concerned with evaluating designers' decisions on (1) whether they needed to study the present organizational history, structure, and culture; (2) if so, what aspects they needed to study; and (3) given the choice of aspects to study, the extent to which they had to be examined. Like designers, auditors will have to consider the context in which these decisions were made and reflect on the choices they would have made themselves. They can then compare the designers' choices against their own to evaluate the implications of the similarities and differences for the conduct of the remainder of the audit.

3.4.4.2 Studying the Existing Product and Information: -

Studying the existing product and information flows is important for three reasons. First, sometimes the design of the proposed system will be based primarily on current product and information flows. The new system is not intended to be a radical departure from the status quo. Second, designers might need to understand the strengths and weaknesses of the existing product and information flows to determine
the new system requirements. In particular, current system weaknesses might have motivated the request for a new system. Third, a good understanding of the existing product and information flows might be required to assess the extent of change needed to implement and operate the new system. Designers can then make better decisions on how to manage the change process.

When designers study the existing product and information flows, auditors should have three concerns. First, they must evaluate the quality of stakeholders' decision about the nature and extent of the examination they chose to make. Auditors must consider whether stakeholders made astute decisions in light of the contextual factors that affect their choice. Second, auditors should examine whether stakeholders used a high-quality methodology such as structured analysis or object-oriented analysis, to guide their examination of the existing product and information flows. Moreover, if designers did employ a formal methodology, auditors must evaluate how well the designers used it. Third auditors should determine whether designers used computer-aided software engineering tools (CASE) to support their analysis and documentation of existing product and information flows. If so, the quality of their work is likely to be higher.

3.4.5 **Formulation Strategic Requirements:**

The strategic requirements for a system specify that overall goals and objectives the system must accomplish. Strategic requirements are identified based on perceived deficiencies in the existing system or perceived opportunities for enhanced task accomplishment and quality of working life.

Socio-technical design theorists stress the importance of carefully eliciting a system's strategic requirements before design work commences. They say that many system failures can be attributed to inadequate performance of this activity. Fundamental difficulties arise because stakeholders do not recognize they have different and sometimes conflicting strategic requirements for the proposed system. Management's strategic requirements, for example, tend to be task-accomplishment oriented. Users' strategic requirements, on the other hand, tend to focus on quality-of-working-life.
issues. The two sets of strategic requirements might be incompatible. If subsequent
collision is to be avoided or reduced, the strategic requirements that ultimately guide
system-design work will have to be negotiated carefully.

Auditors should be concerned to see that system designers recognize the importance
of articulating strategic requirements for the quality of subsequent design work. If the
proposed system will have a substantial behavioral impact, they should examine and
evaluate the procedures used by the stakeholders to reach agreement on strategic
requirements. If substantial uncertainty surrounds the proposed system, they should
examine and evaluate the procedures used to help clarify strategic requirements.

3.4.6 Organizational and Job Design:

In some cases, achieving the strategic requirements chosen for a system will
necessitate the initial design or redesign of organizational structures and jobs. In
choosing organizational structures and job designs for those parts of the organization
that will be affected by the proposed system, many factors should be considered. For
example, if substantial uncertainty surrounds the tasks to be accomplished in the
proposed system, loose, organic organizational structures and job designs might be
successful. These types of organizational structures and job designs promote
creativity, innovation and free flows of information needed to address the stresses and
uncertainties that can undermine task performance and quality of working life.

The design of organizational structures and jobs can be a complex activity. If auditors
assess that a proposed system will impact organizational structures and jobs, they
would be concerned to see the systems development team obtained high-quality
advice from someone skilled in organizational theory and practice. They would seek
evidence, for example, on whether personnel assigned responsibility for
organizational structure and job design contained representatives of stakeholder
groups, how the design tasks were undertaken, the processes used to resolve conflict
and uncertainties and the level of consensus achieved in relation to the designs finally
chosen. If auditors conclude that these types of issues have been resolved
satisfactorily, they can reduce the level of control risk associated with systems
development. If the issues have not been resolved satisfactorily, however, auditors must revise their assessment of control risk upward and plan increased substantive testing accordingly.

3.4.7 **Information Processing Systems Design:**

In the systems development process, the information processing systems design phase is one of major involvement. From a systems effectiveness viewpoint, the auditor considers whether the design meets the strategic requirements agreed upon by the stakeholders. From an efficiency viewpoint, the auditor assesses the reasonableness of the resources required to operate the system. From an asset safeguarding and data integrity viewpoint, the auditor evaluates the likely reliability of the controls designed into the system.

During this phase, auditors should also evaluate the auditability of the system. They might deem it necessary, for example, to build certain audit capabilities into the system in the form of audit modules. These modules capture data or examine conditions of interest concurrently with production running of the system. Auditors might also deem it necessary to purchase certain tools in order to audit the system. When “evaluating the information processing systems design phase, either as a participant in the design process or in a post implementation or general ex post review capacity,” auditors must examine the conduct of six major activities.

1. Elicitation of detailed requirements,
2. Design of the data/information flow,
3. Design of the database,
4. Design of the user interface,
5. Physical design, and
6. Design and acquisition of the hardware/system software platform.
As with the previous phases, these activities can vary considerably depending on the type of system being designed and implemented. For example, compared with a large, critical application system, they will be less onerous in a small end-user-developed application system.

3.4.8 **Application Software Acquisition and Development:**

After the information processing systems design phase is complete, application software might have to be acquired or developed. In some cases generalized software packages can be purchased to perform all or some functions within the system. For example, an order-entry package could be purchased to perform sales functions or a database management package could be acquired to assist with control and use of data. These packages have to be configured and perhaps modified and adapted. In other cases, the system might exist in a prototype form at the conclusion of the design phase. Work might be undertaken to tune the prototype so it runs more efficiently or to write the prototype in another programming language that will execute more efficiently. In still other cases, new programs must be developed from scratch. Various activities are involved: design, coding, compiling, testing and documenting.

Auditors might have several concerns during the application software acquisition and development phase. If application software is acquired, they should be concerned about the adequacy of the requirements specification provided to vendors, the quality of the procedures used to evaluate the software tendered in terms of such factors as functionality, accuracy and completeness, quality of documentation, vendor stability and support and the nature of the terms and conditions in the contract exchanged with the vendor. They must evaluate the likely quality of the application software acquired and the adequacy of vendor maintenance and support.

If, on the other hand, application software is developed, auditors must be concerned about the procedures undertaken during the design, coding, compiling, testing and documentation phases. If software development is undertaken by information systems professionals, auditors usually can assess control risk as lower. Skilled programmers, for example, should be aware of the quality-assurance procedures they must follow to
reduce the number of errors in the programs they write. On the other hand, if end users are developing material systems, auditors often must assess control risk as higher. In spite of the criticality of systems they sometimes develop, many end users lack knowledge of and experience with the quality assurance procedures needed to produce high-quality programs.

During this phase, auditors could be involved in acquiring or developing software for their own purposes. For example, they might wish to embed certain audit modules in the system so they can continuously monitor the system. They might also seek to acquire generalized software to help audit the system. Auditors must be vigilant in applying the same quality assurance measures expected of the persons audited.

3.4.9 **Hardware/System Software Acquisition:**

If new hardware or system software must be purchased to support the new application system, a request for a proposal must be prepared. Vendor submissions must then be solicited. These submissions must be evaluated and a final selection made.

Auditors have similar concerns during this phase as they do during the application software acquisition and development phase. Auditors should seek evidence that designers carefully prepared a request for proposal for any new hardware/system software needs. The request for proposal should document such matters as transaction volumes, database sizes, turnaround and response time requirements, and vendor support needed. They should evaluate how designers used the request for proposal to assess vendor submissions. For example, auditors should evaluate how well the hardware and system software finally chosen enable the overall design goals of modularity and generality to be achieved. Finally, they should be concerned about such matters as vendor viability, ongoing support, the availability of source code and maintenance support in the event of vendor failure and contractual obligations to the vendor.
3.4.10 **Procedures Development:**

During the procedures development phase, designers specify the activities that users must undertake to support the ongoing operation of the system and to obtain useful output. In general, the objective should be to provide minimum specification of procedures. Inevitably, many users will disregard instructions and try to develop their own approach to accomplishing a task. From the viewpoint of maintaining control, what needs to be done should be clearly specified. Where possible, how it should be done should be left to the person responsible for the task. Auditors might provide suggestions, but they must accept that users might not follow these suggestions.

Procedures development involves four major tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Design of procedures</strong></td>
<td>To the extent procedures must be specified, they must be matched with the job/task design. What triggers the task and the task input and output must be identified.</td>
</tr>
<tr>
<td><strong>Testing of procedures</strong></td>
<td>Users and operators must test the adequacy of the procedures design. They might suggest modifications in light of their experience.</td>
</tr>
<tr>
<td><strong>Implementation of procedures</strong></td>
<td>Conformity with system procedures represents the most direct way people have to change their behavior when a new system is implemented. Where systems have a substantial behavioural impact, management of the change process can be especially critical during the conduct of this task.</td>
</tr>
<tr>
<td><strong>Documentation of procedures</strong></td>
<td>User and operator procedures must be documented formally. Where possible,</td>
</tr>
</tbody>
</table>

72
procedures manuals should be written in a consistent, formal style.

Auditors should have several concerns about the conduct of the procedures development phase. First, they must assess the quality of the procedures design. In particular, they should evaluate whether the principle of minimum specification of procedures has been observed. Second, if the system will have or has had a substantial behavioral impact, they should check to see the procedures design team contained representatives of all important stakeholder groups. Third, auditors should evaluate the approach used to test procedures. The wider the impact of the system, the more critical it is that procedures be tested thoroughly before they are implemented operationally. Fourth, they should evaluate the quality of procedure documentation. Again, documentation becomes increasingly important as the system has a wider impact.

3.4.11 Acceptance Testing:

The purpose of acceptance testing is to identify as far as possible any errors and deficiencies in the system prior to its final release into production use. Errors and deficiencies can exist in the software supporting the system, the user interface, procedures manuals, job design, organizational structure design and so on. Acceptance testing is carried out to identify these errors or deficiencies before they have a widespread impact.

The conduct of acceptance testing can vary considerably, depending on the type of system being implemented and the activities undertaken during systems development. For example, if substantial amounts of program code have been written, acceptance testing must involve ensuring the code is authorized, accurate and complete. If generalized software has been purchased and the software has an extensive user base, acceptance testing might focus primarily on job designs or user procedures associated with the system. If the system has been developed using a high-level language, testing might be iterative as various system prototypes are tried and either enhanced or discarded.
For several reasons, acceptance testing usually cannot be comprehensive; that is, all system features cannot be tested. First, it is difficult to conceive of every execution path through a system of even moderate complexity. Second, deficiencies in a system might become apparent only after extensive experience with the system. Third, it is difficult to conceive of every condition under which the system must operate. Exceptional circumstances could arise, such as an abnormal system load or a rare combination of transactions, which were not anticipated in the design of acceptance tests. Fourth, in some cases it is difficult to know whether a result is correct, anyway. Fifth, even with the availability of testing tools such as test data generators, acceptance testing can be expensive. At some point the costs of extra testing outweigh the expected benefits. Given these difficulties, information systems professionals now try to design and implement systems without errors or deficiencies in the first place rather than attempt to remove errors during acceptance testing.

<table>
<thead>
<tr>
<th>Type of Testing</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program testing</td>
<td>Programmers who develop the individual programs in the system must test their programs for accuracy, completeness and efficiency.</td>
</tr>
<tr>
<td>System testing</td>
<td>Some members of the development team must take responsibility for testing the overall system to see especially that the interfaces between the various programs and subsystems work correctly.</td>
</tr>
<tr>
<td>User testing</td>
<td>Users must test the total system, including the organizational structure design, the job design, system interfaces, programs and procedures.</td>
</tr>
</tbody>
</table>
Quality assurance testing

A quality assurance group is responsible for ensuring the system complies with all standards adopted by the organization.

Auditors should seek to answer the following types of questions when evaluating the conduct of work carried out during the acceptance testing phase:

- How was the testing process planned?
- How were test data designed and developed?
- What test data were used?
- What test results were obtained?
- What actions were taken as a result of errors or deficiencies identified?
- What subsequent modifications to test data were made in light of testing experience?
- How was control exercised over test data and the acceptance testing process?

Auditors usually strive to obtain high-quality answers to these questions. The way in which acceptance testing is undertaken often has a major bearing on their estimates of the control risk associated with the systems development management subsystem. Good acceptance testing procedures force high quality work to be undertaken in the other phases of systems development. If acceptance testing procedures are weak, however, controls elsewhere in the systems development process tend to deteriorate.

3.4.12 Conversion:

The conversion phase comprises those activities undertaken to place the new system in operation. In some cases the transition must, be made from an existing system. In other cases the system has no predecessor. In all cases, however, conversion requires users to adopt new behaviors. Accordingly, it is often a critical phase within the overall management of the change process.
Depending on the nature of the system being developed and implemented, conversion could be a minor step or it could involve major efforts over an extended period. If some type of system exists already, conversion to a new system can occur in one of three ways. First, the old system could be stopped abruptly to make way for the new system. This strategy reduces the costs of conversion because no redundant processing occurs. The costs of any deficiencies discovered in the new system can be high, however, as the old system is not available as backup. Second, both systems could run in parallel for a period (but performing different functions) with users employing output from both systems. Again, this phase-in conversion strategy reduces conversion costs. Moreover, it permits orderly changeover to the new system. Nevertheless, as with the first strategy, the costs of deficiencies in the new system can be high because no backup is available. Users might also encounter difficulties in having to work with two systems. Third, both systems could run in parallel (performing the same functions) with either the old system output or the new system-output being used. In this last case, parallel running provides the basis for validating the design and implementation of the new system. Thus, this strategy reduces the risks associated with conversion. Redundant processing could be costly, however and users might also encounter difficulties in having to work with both systems.

Auditors often pay special attention to several aspects of the conversion phase:

1. If substantial disruption is likely to occur, asset safeguarding, data integrity, system effectiveness and system efficiency are at risk. For example, a programmer could take advantage of a situation in which managers have insufficient time to review program modifications to install unauthorized code. Likewise, a data-entry clerk could introduce unauthorized transactions into the system when large backlogs of input exist and many data-entry errors have to be corrected.

2. Conversion can be a time when tempers fray and users become severely disillusioned with the system. As a result, they might begin to undermine implementation efforts. Careful management of the change process becomes especially critical when users engage in counter implementation activities.
3. Often trade-offs must be made between the integrity of data taken up on the new system and the need to get the system running. For example, data validation criteria might be relaxed because high rejection rates are encountered and the conversion process is stalling. Later correction of data must not be forgotten.

4. Careful planning of the activities to be undertaken during the conversion phase is essential for many systems. Controls to ensure asset safeguarding, data integrity, system effectiveness and system efficiency must be designed and implemented. For example, control totals might be used to ensure data converted from one storage medium to another is not corrupted or lost. The conversion phase also must be monitored carefully to identify problems and undertake remedial actions promptly.

3.4.13 Operation and Maintenance:

During the operation and maintenance phase, the new system is run as a production system. In addition, periodically it is modified to better meet its objectives. Only through day-to-day experience with a system do many of its strengths and weaknesses become apparent. In light of production experience with a system, three types of changes can be undertaken: (1) repair maintenance—logic errors discovered in the system are corrected; (2) adaptive maintenance—changes in the system (user) environment might necessitate system modification and (3) maintenance changes might be made to improve processing efficiency. Repair maintenance and perfective maintenance are more likely to occur early in the life of a system. Adaptive maintenance is more likely to occur at later stages.

Whatever the reason for maintenance, auditors' primary concern is that a formal change process exists to identify and record the need for changes to a system and to authorize and control the implementation of needed changes (Dow and Gallegos 1986). This formal change process is more important in systems that are used widely throughout the organization and which perform basic and critical functions. Nevertheless, even in small systems developed by users for their own purposes, formal change procedures can still be important. Small, localized systems could support users
who are making critical decisions affecting the overall welfare of the organization. The introduction of erroneous modifications into these systems can have widespread consequences. In this light, maintenance activities associated with these systems need to be approved and monitored carefully.

3.5 **Information System Auditor's Role in System Development:**

Auditor involvement may vary from project to project and auditors may not be involved in every systems development project. Each system development project will need to be risk assessed to determine the level of audit's involvement. The type of review will also vary depending upon the risks of particular project. Auditors may only be involved in key areas or the entire development project. In any case, auditors need to understand the development process and application controls to add value and ensure adequate controls are built into the system. Auditors can take on two different roles in a systems development project: control consultant or independent reviewer. As a control consultant, the auditor becomes a member of the development team and works with analysts and programmers to design application controls. In this role, the auditor is no longer independent of the development team. As an independent reviewer, the auditor should have no design responsibilities and not report to the project team, but can provide recommendations to be acted on or not by the project manager.

The scope of a system development audits can include an evaluation of the software development life cycle or an evaluation of the quality of the deliverables from each system development and implementation phase. Recommendations from systems development audits might include improvements in user requirements, application controls, or the need to document test plans and expected test results. Developing new systems can be costly and time-consuming endeavor. A well-controlled environment with overall strategy, standards, policies, and procedures helps ensure the success of development efforts. There are many processes that need to be well controlled to ensure the overall success of an application: analysis, design, testing, and implementation. Because the cost to implement controls after a system has already gone into production, controls should be defined before a system is built.
There are many opportunities for auditor involvement in the development process. Auditors need to develop the skills and relationships to work with the development team to ensure that controls are built into the system. Auditors can assist organizations by reviewing the systems development environment, evaluating standards for systems development, and evaluating phases in the systems development process. Auditors can assist management by reviewing critical systems for input, processing, and output and verifying that the new system provides an adequate audit trail. The auditors role in a systems development project depends on the organization’s culture, maturity of the IS functions, and the philosophy of the auditing department.

Auditing systems development requires specific knowledge about the development process and application controls. Understanding the development process allows the auditor to identify key areas that would benefit from independent verification. Understanding application controls allows the auditor to evaluate and recommend controls to ensure complete and accurate transaction processing. By becoming involved at strategic points, the auditor can ensure that a system is well controlled and auditable. The following list highlights some of the key tasks the auditor may perform during a system’s development:

- Review user requirements
- Review manual and application controls
- Check all technical specifications for compliance with company standards
- Perform design walkthroughs at the end of each development phase
- Submit written recommendations for approval after each walk-through
- Ensure implementation of recommendations before beginning the next phase
- Review test plans
- Present findings to management
- Maintain independence to remain objective
These tasks can help minimize control weakness and problems before the system becomes operational rather than after it is in use. To determine the level of involvement, the auditor should first complete the risk assessment of the systems development process and determine the amount of time to allocate to a particular development project. Next, the auditor should develop an audit plan that includes a schedule for the specific review points tied to the development schedule. Finally, the auditor needs to communicate the scope of involvement and any findings to development, users, and management.

References:


5. www.acl.com

6. www.cit.icai.org

7. www.icai.org

8. www.idea.com

9. www.isaca.org