CHAPTER 03
CONTROL OBJECTIVES

1. ***CONTRO***L

Checks or management tools which are implemented to ensure that process or system willwork as per its intended purpose

Types of control systems:

System development & acquisition controls
Control over system implementation
Control over system & program changes
Application controls
Logical access controls
Physical access controls
Environmental controls

Classification controls:

General Controls: those controls which are applicable to overall system components, processes & data for a given organization or systems environment. Include data centers & networking operations, system development & acquisition, system change & maintenance access & computer processing.

Application Controls: those controls that are applicable to individual accounting subsystems such as payroll or accounting payable.

2. IMPORTANCE OF CONTROL

Information is an important resource
Increasing threats of various types to IS
Increasing need for regulatory compliance
IS is a set of integrated resources
Growing importance, education & awareness of information security & controls

3. EFFECTS OF COMPUTERS ON INTERNAL AUDIT

Finance & accounting system has become more automated
Changed the ways auditors used to carry out auditing
Changed audit style from manual audit to ***audit around computers***, & then to ***audit through computers***

Impact of computers in preparation & maintenance of financial books & reports:

Changes in audit trail & audit evidence:
Reasons of changes in audit trail in computerized IS compared to manual system:

Data retention & storage
Absence of input documents
Lack of visible audit trail
Lack of visible output
Audit evidences
Legal issues

Changes in internal control environment:
Following are used in both manual & computerized systems:

Personnel
Segregation of duties
Authorization procedures
Record-keeping
Access to assets & controls
Management & supervision & review

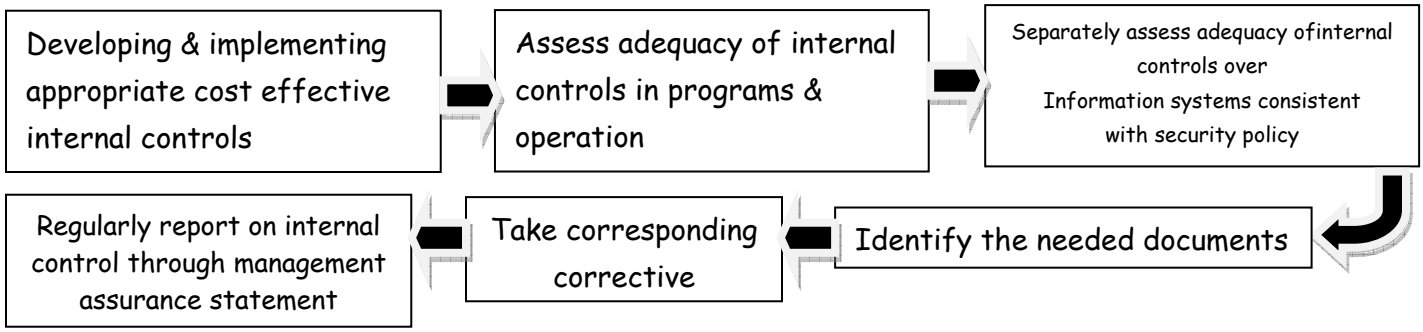
Following is the difference between manual & computerized systems:

Concentration of programs & data

- New causes & sources of error:
  System-generated transactions
  Systematic errors

- New audit procedures:

4. RESPONSIBILITIES FOR IMPLEMENTING CONTROLS



5. CONTROL OBJECTIVES FOR INFORMATION RELATED TECHNOLOGY (COBIT)

COBIT is a set of best practices or framework for IT management created by **Information Systems Audit & Control Association (ISACA)** & **IT Governance Institute (ITGI)** in **1996.**

COBIT provides framework for:

- Management
- Users
- Auditors

COBIT provides a set of generally accepted indicators, processes & best practices to assist them in maximizing benefits derived through use of IT
It helps in appropriate IT governance & controls in a company.
Provides balance between risk & control investment in the IS environment

COBIT framework addresses issues of controls effectiveness from three dimensions:

- Business objectives
- IT sources
- IT processes

6. IS CONTROL TECHNIQUES

Control framework primarily divides controls into 3 categories:

- Accounting controls
- Operational controls
- Administrative controls

COSO's Objectives for organizations:

- Reliability of financial reporting
- Effectiveness & efficiency of operations
- Compliance with applicable laws & regulations

Auditor's categorization of controls:

Preventive Controls: designed to an error or malicious activity in system. Devised by understanding probable threat, understanding vulnerabilities & exposure of assets for threats & finding necessary preventive controls to avoid probable threats.

Detective Controls: used to detect errors or malicious activities in system. Devised by having clear understanding of lawful activities, using preventive controls & establishing detective controls which report unlawful activities.

Corrective Controls: designed to reduce impact of errors or malicious activities by correcting errors & avoiding malicious activities in future. Help to minimize impact of threats or problems, rectify issues & modify processing system to minimize future occurrence of problems.

Compensatory Controls: where organizations can't implement preventive controls, compensatory controls are used.

7. AUDIT TRAILS

Audit trails used as detective controls.
Logs that can designed to record user activities on system & applications.
Help to accomplish security policy
Objectives of audit trail:
Detecting unauthorized access to system
Reconstruction of events
Personal accountability

8. SYSTEM DEVELOPMENT & ACQUISITION CONTROLS

There are 3 major threats to system development & acquisition:

May consume excessive time & resources
May not be developed as per requirement
May contain unauthorized/ fraudulent instructions

Following components can be uses to protect system from above problems:

Developing strategic master plan
Use project control techniques:
System authorization
User involvements
Technical design
Testing
Internal auditor's involvement
User acceptance & testing
Proper data processing schedules
Measuring system performance:
Response time
Utilization
Stress test
Throughput
Post-implementation review
Post-implementation review

9. CONTROLS OVER SYSTEM IMPLEMENTATION

Once system is developed & ready, User Acceptance Test is carried out. Its aim is to Confirm that:

- User requirement specifications are met
- Operational documentation is accurate, comprehensive & usable
- Helpdesk & other ancillary functions are properly operating
- Back-up & recovery procedures will work effectively

Acceptance testing includes:

- Performance testing
- Volume testing
- Stress testing
- Security testing
- Procedure testing
- Back-up & recovery
- Parallel operation

Auditor's validations for user acceptance & testing:

Auditor has to ensure & verify that:

- An acceptance test plan has been drawn up
- A manager with adequate authority has been appointed to handle proceedings
- Testing is working in actual environments
- Proper data conversion plan has been developed considering all material issues
- Adequate controls exist to prevent unauthorized changes from being made
- Proper resources are available
- Responsibility allocation is complete
- Ensure that end-users are fully involved

Post-implementation review (PIR):

PIR must evaluate whether the implemented system has met its:

- Business objectives
- User expectations
- Technical requirements

10. CONTROLS OVER SYSTEM & PROGRAM CHANGES

Following controls should be used for control over system & program changes:
Change management controls
- Authorization controls
- Documentation controls
- Testing & Quality controls

11. CONTROLS OVER DATA INTEGRITY & SECURITY

Information classification is done as per level of sensitivity of information. Information is to be classified under:

- Top secret
- Highly confidential
- Proprietary
- 'for internal use'
- Public documents

Data integrity aims to prevent, detect & correct errors in transactions as they flow through various stages of data processing. Further, data integrity helps protect data from malicious or accidental data alteration or destruction & provide assurance about quality & integrity of information to the users.

There are 6 data integrity controls:
- Source data controls
- Input validation routines
- Online data entry controls
- Data processing & storage controls
- Output controls
- Data transmission controls

12. ACCESS CONTROLS

IS has two types of access:
- Logical access
- Physical access

13. LOGICAL ACCESS CONTROLS (LAC)

Objectives:

- Only Authorized access to system
- Restrict users to only authorized transactions
- Restricted access of network to authorized only
- Protecting system from malicious programs & viruses
- Helps to protect integrity of application & data

Types of Logical Access Paths:

- Online terminals
- Operation console (directly connected to servers)
- Dial-up ports
- Telecommunication links
- Batch processing

Possible exposures/ revelations in losses:

- Technical exposures
- Asynchronous exposures (attacks)
- Loss of exposures

LAC violators:

- Hackers
- IS Personnel
- End users
- Former employees
- Interested or educated outsiders
- Competitors & crackers
- Part-time & temporary personnel

Types of LAC:

- Using User-id & password
- Using access control
- Using data encryption
- Using firewall
- Using network monitoring

Access of Control mechanisms:

Identification & authentication
Authorization

Audit of LAC: Role of IS auditor:

- Review relevant documents relating logical access & associated risks
- Review potential unauthorized access paths
- Review working of various logical access controls
- Review password policy of organization

14. PHYSICAL ACCESS CONTROLS

Reasons of physical access violations:

- Abuse of data processing resources
- Embezzlement or fraud
- Blackmailing or revenge
- Damage to equipments & resources
- Theft of equipments & resources
- Public disclosure of sensitive information

Physical access violations are mostly done by:

- Ignorant employees
- Former employees
- Striking employees
- Interested or informed outsiders

Audit of physical access controls:

Assess various threats & risks to facilities
Review controls used to avoid these threats & risks
Observe & test controls to protect hardware facilities, computer terminals

15. ENVIRONMENTAL CONTROLS

Primarily due to elements of nature.

Common environmental threats are:

- Fire damage
- Water damage/ flooding
- Power strike
- Electrical shock

Natural disasters
Equipment failure
AC failure
Bomb threat/ attack

16. SECURITY CONCEPTS & TECHNIQUES

Cryptosystems
Data encryption standards
Private key encryptions
Public key infrastructure
Firewall

CHAPTER 04
TESTING – GENERAL & AUTOMATED CONTROLS

Testing is a process to verify correctness, completeness & quality of developed software or any other product. It is known as criticism or comparison. It helps to verify that developed software or product would be working, as it is intended to.

1. SOFTWARE TESTING FUNDAMENTALS

Following rules to be followed as software testing fundamentals:

Testing objectives:

- Program with an intention to find errors
- Having high probability of finding yet undiscovered errors
- Uncovering a yet undiscovered error

Starting time of test:

- Should start as early as possible in life cycle as early testing helps to reduce errors

Cause of bugs:

- Specifications
- Design
- Coding

Testing costs:

- Costs increase manifold if bugs not detected on time. Fixing bugs at early stages of life cycle will cost less.

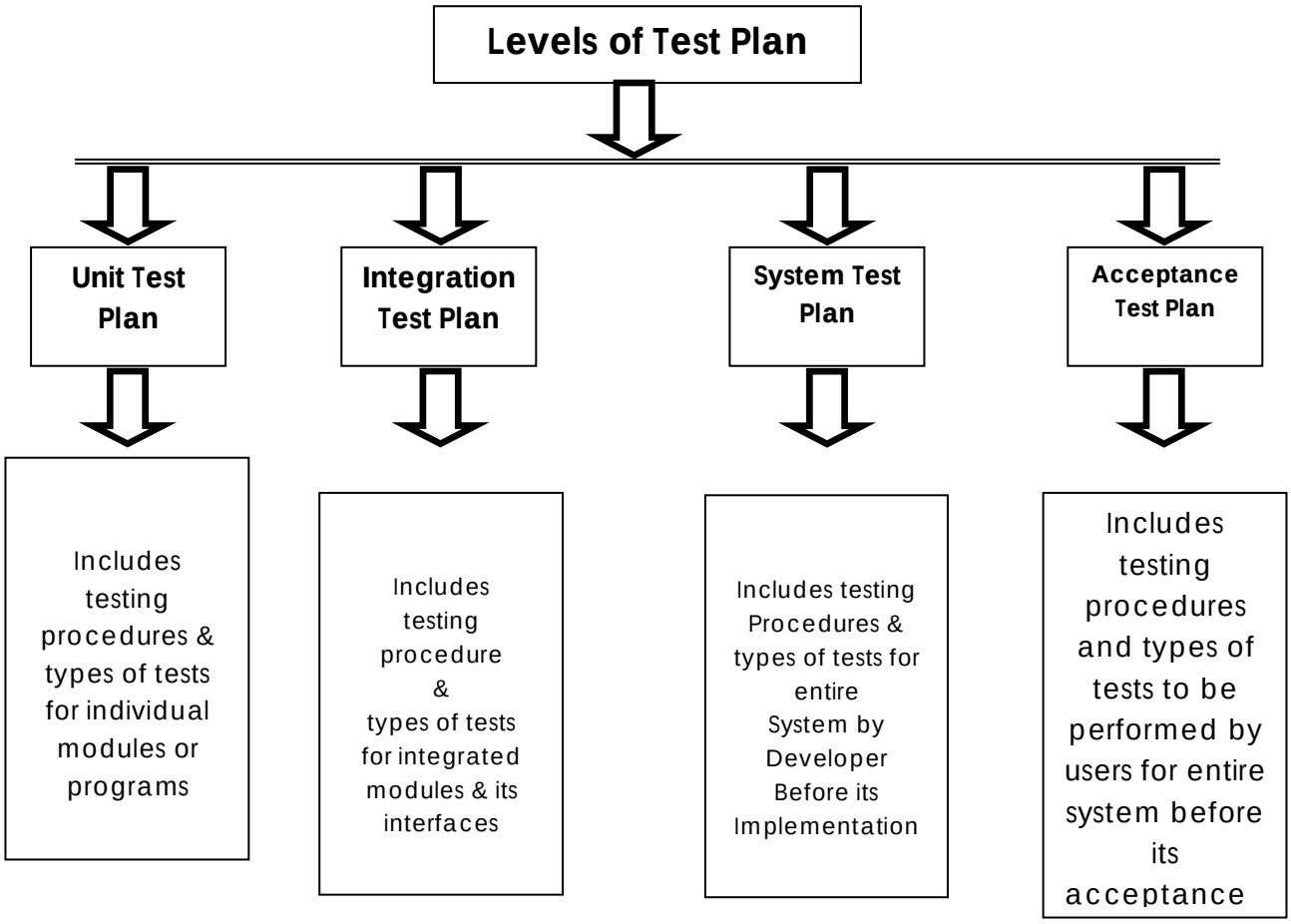
Stopping time of testing:

- When deadlines are met
- When test cases are completed
- When testing budget is depleted
- Rate of finding bugs is too small
- Risk in project found to be under acceptable limits

Testing strategy:

- Specific
- Practical justified

2. LEVELS OF TEST PLAN



3. TEST PLAN OUTLINE

a) Background
b) Introduction
c) Assumptions
d) Test items
e) Features to be tested
f) Features not to be tested
g) Approach
h) Test item pass/ fail criteria
i) Suspension/ resumption criteria
j) Test deliverables
k) Environmental needs
l) Responsibilities
m) Staffing & training
n) Schedule
o) Resources
p) Risks & contingencies
q) Approvals

4. TYPES OF SOFTWARE TESTING

- **Static testing:** used for verification activities. To check whether work being done is as per is as per set standards of organization.
- **Dynamic testing:** involves working with software, giving input values & checking if output is as expected. Includes validation activities, unit tests etc.

5. BLACK BOX TESTING

Treats software as a “black box, without any knowledge of internal implementation”
Valid & invalid inputs selected to determine accuracy of outputs
Mainly concerned with correct acceptance of inputs & relevant outputs
Applicable to all levels of testing
Can cover unimplemented parts also
Testing method includes:

- Equivalence partitioning
- Boundary value analysis
- Cause effect graphics techniques

Equivalence Partitioning: software testing technique that divides input data of software unit into partition of data from which test cases can be derived. Tries to define test cases that uncover classes of errors. Its goals are:
1) to reduce number of test cases to necessary minimum &
2) to select right test cases to cover all possible scenarios.
Boundary Value Analysis (BVA): since more application errors occur at boundary levels of input domain, BVA is used to identify errors at boundaries rather than finding them in center of input domain.
Cause Effect Graphing Techniques (CEG): provides a concise representation of logical conditions & corresponding actions. Performed once requirements have been reviewed for ambiguity & content is reviewed. There are 4 steps if CEG: Causes & effects listed for a function Development of cause-effect graph Graph converted into decision table Decision table rules converted to test cases

6. WHITE BOX TESTING

Used when tester has access to internal data structure & algorithms & also access to implement these algorithms
Provides test cases for testing internal structure of programs
Used to test all details of software for:

- Logical errors &
- Syntax errors (typographical errors)

Types of tests used:
Basis path testing: every path of program is derived & tested. Ensures that every statement is executed & tested at least once, takes help of flow graphs to simplify derivations of each path.
Loop testing: focuses exclusively on validity of loop constructs. There are four different classes of loops: *simple loop, nested loop, concatenated loop & unconstructed loop.*

7. UNIT TESTING

Software verification & validation method where programmer gains confidence that individual units of source code are fit for use.
Done by stepping through each & every line of code.
A unit test is a method of testing the correctness of a particular module of source code & write test cases for every function in module such that each test case is separate from others.

Benefits:

Encourages change
Simplifies integration
Documents the code

Limitations:

Cannot capture every error in program
Cannot catch integration errors

8. REQUIREMENT TESTING

Helps to ensure that system's requirements communicated by users are correct & requirements are correctly understood & specified by developers for further system development.
Helps in developing correct system in efficient manner

Objectives:

To ensure that system's requirements are correctly communicated & specified
To ensure that system performs correctly for sustainable & considerable period of time
To ensure successful implementation of user requirements
To ensure that all user needs are fulfilled
To ensure compliance organization's policies & procedures

Advantage: minimizing extensive re-work by minimizing requirements-related defects that could have been prevented/ rectified earlier.

9. REGRESSION TESTING

"The process of testing changes to computer programs to make sure that older programs still work with new changes."

- A regression test re-runs previous tests against changed software to ensure that changes made in current software do not affect functionality of existing software.

- **Objectives:**
  - To ensure that older programs still work with new changes
  - To ensure that documents remain updated as per new software
  - To ensure that test-data & test-conditions remain updated as per changes

- To be used when there is high risk that new changes may affect unchanged areas of application software.
- Used in both development & maintenance phases of software.

10. ERROR HANDLING TESTING

ERROR HANDLING refers to the detection and resolution of programming, application and communication errors. There are 2 types of errors:

a. Development error (also known as syntax and logical error) – Happens due to typing mistakes or incomplete logics; and
b. Run time error – Happens due to invalid inputs

Error handling testing means establishing that all error handling procedures are in place for applications, i.e. errors will be detected and handled as per set guidelines.

Error handling Objectives are:

Applications recognizes all expected error conditions,
High probability that errors will be corrected, and
Reasonable controls will be maintained for error corrections

11. MANUAL SUPPORT TESTING

MANUAL SUPPORT means that certain functions will be performed by people rather than automated IS. Manual Support Testing is the testing of the manual support functions. Manual support testing is best done during the installation phase. It has the following objectives:

Verify that manual support procedures are correct and documented properly
Determine that manual support responsibilities are established & defined correctly

Determine that manual support people are adequately trained
Determine that manual support procedures are properly connected

12. INTER SYSTEM TESTING

Inter System Testing ensures that interconnection between various systems & applications work properly. It is done to ensure that:

Parameters and data are correctly passed between linked applications and systems
Proper coordination between working of different linked systems exists

13. CONTROL TESTING

Control is a check and it acts as a management tool to ensure that processing is performed in accordance to the intents of the management. It ensures that:

Input data is accurate & complete
Transactions being processed are authorized
Audit trail is in existence
Processes being used are efficient, effective & economical
Processing meets the needs of the users

Methods to use control testing
First, all risks are to be identified
Testers should have negative approach
Testers should run test data and ensure that controls are effective

14. PARALLEL TESTING

It is done to ensure that the processing of the new application is consistent with respect to the processing of the previous system.
The objective is to ensure that the new system performs correctly and to demonstrate the consistency and inconsistency between the 2 applications.

15. VOLUME TESTING

It is the testing of the system when the maximum numbers of users are simultaneously active and when the database contains the greatest volume of data.
The purpose of volume testing is to find out the weakness in the system with respect to its handling of large amount of data during the extended time periods.

16. STRESS TESTING

The purpose of stress testing if to find defects in the system capacity of handling large numbers of transactions during peak periods.
Server testing deals with the quality of the application in the expected environment. The idea is to create a environment more demanding of the application than the one the application would experience during its normal course of operation.

17. PERFORMANCE TESTING

System performance is generally assessed in terms of response time and throughput rates under different processing and configuration conditions.
The performance problems are most often the result of the client or server being configured inappropriately.

The best strategy for improving client server performance is a three step process:

1st, execute controlled performance tests that collects data about volume, stress and loading tests.
2nd, analyse the collected data
3rd, examine and tune database queries.

18. CONTINUOUS AUDIT TECHNIQUES

Integrated Test Facility (ITF):

Used in IS which is to be audited
Auditor uses a dummy account for testing & reviewing. The dummy transactions do not affect IS.
Mainly used in online system
Test transactions can be submitted along with actual transactions on frequent basis without disrupting regular processing operations.

Snapshot Technique:

Involves having audit software taking pictures of transactions as it flows through an application system.
Both before-images & after-images of transactions are captured
Decision to be made regarding location of snapshot points & reporting of snapshot data captured

System Control Audit Review File (SCARF):

Similar to snapshot technique
An embedded audit module is used to continuously monitor transactions & collect data on transactions with special audit significance
One of most complex online audit techniques
The audit modules are placed at predetermined points to gather information about transactions which auditors deem to be material
Two key decisions: what information to be collected & what reporting system to be used.

Continuous & Internal Simulation (CIS):

Embeds audit module in DBMS
This can be used when application system uses DBMS
Uses DBMS to track exceptional transactions

Advantages of continuous audit:

Timely audit
Comprehensive & detailed auditing
Surprise test capability
Assessing whether IS meets set objectives
Training for new users

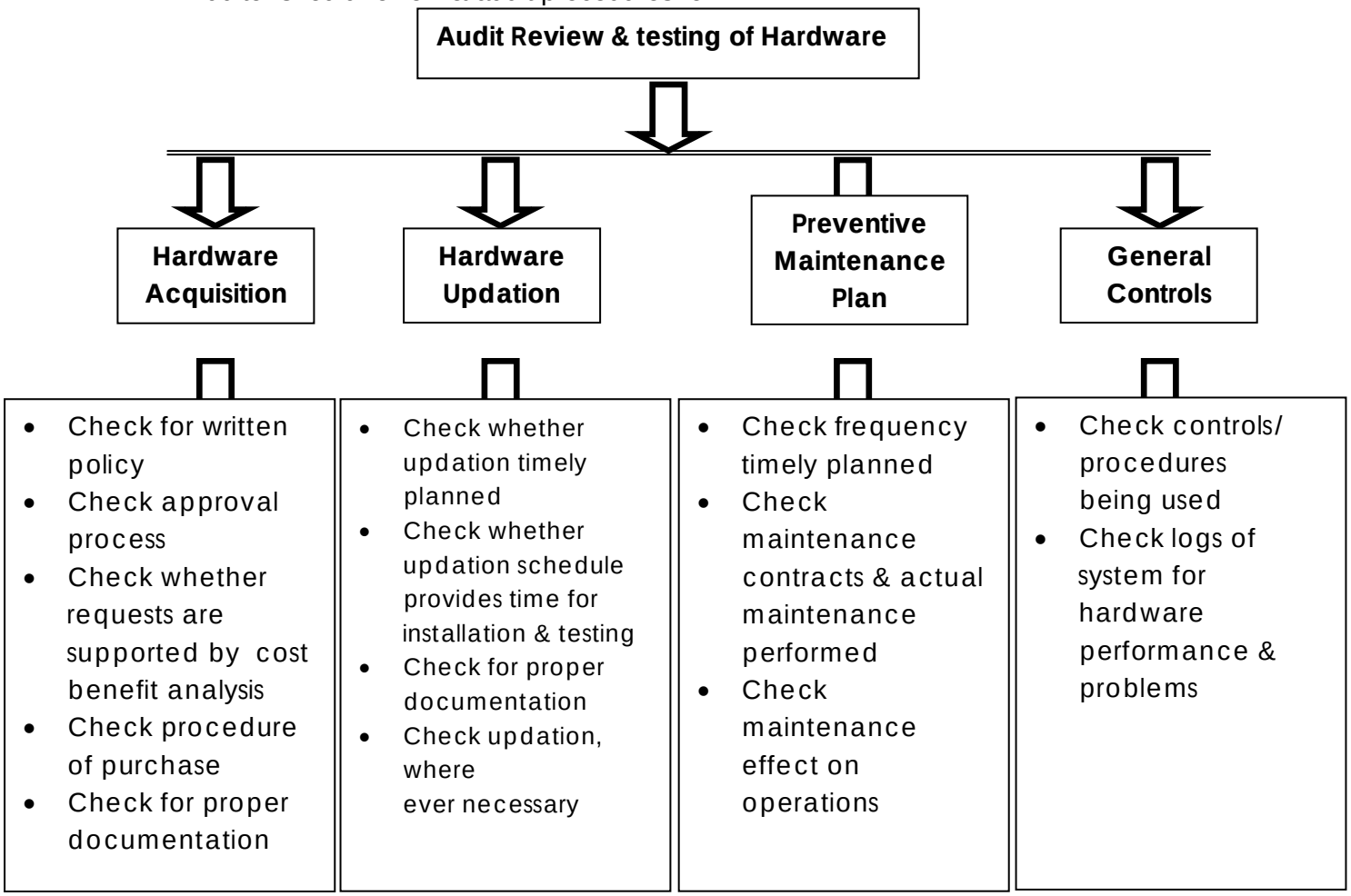
Disadvantages of continuous audit:

Rationing of available resources
Involvement in system development
Expert knowledge of IS being used
Missing audit trail
Stable application system required for implementation

19. HARDWARE TESTING & REVIEW

Normally, hardware should be tested for:
Memory
Performance
Security
Reliability
Error handling/ exit testing
Max. no. of users supportable
Maintenance support
Accessibility testing

Auditor should review & audit procedures for:



20. OPERATING SYSTEM REVIEW

Here, the auditor reviews the procurement, implementation, execution and maintenance of system software such as operating system, in terms of:

Approval process of software selection
Cost / benefit analysis of system software procurement
Controls over installation of the software
System documentation
Test software implementation
System software security procedures

21. NETWORK REVIEW

Network Audit Objectives
Standards are in place for designing and selecting a LAN architecture
Controls are there to ensure continuous working of LAN
Ensure that there is cost benefit in operating a LAN

The reviewer / auditor of networks should have knowledge about networks, network topology,
LAN technicalities etc. The auditor should review, test and validate the following controls for networks:

Physical controls
Logical controls
Environment controls