NABL 137

National Accreditation Board for Testing and Calibration Laboratories (NABL)

SPECIFIC CRITERIA FOR ACCREDITATION OF SOFTWARE & IT SYSTEM TESTING LABORATORIES

Issue No.: 01
Issue Date: 14-Oct-2019

Amendment No.: 02
Amendment Date: 28-Apr-2026

AMENDMENT SHEET

Sl.	Amendment No	Page No.	Clause No.	Date of Amendment	Amendment	Reasons	Signature QA Team	Signature of Competent Authority
1.	02	4, 6, 7	-	10.02.2020	As highlighted	Internal review	-Sd/-	-Sd/-
		10	8.2.1		Inclusion of word 'testing'			
		15	9		Deleted standard name and as highlighted			
2.	02	04 to 19	----	28.04.2026	As highlighted	Internal Review	-Sd/-	-Sd/-
3.								
4.								
5.								
6.								
7.								
8.								
9.								
10.								

CONTENTS

ABBREVIATIONS

APAC	-	Asia Pacific Accreditation Cooperation
CAB	-	Conformity Assessment Body
CCCS	-	Common Control Criteria Standard
CIS	-	Centre for Internet Security
CWE	-	Common Weakness Enumeration
FRS	-	Functional Requirements Selection
GIGW	-	Guidelines for Indian Government Websites
IAAS	-	Infrastructure as a service
IEC	-	International Electro technical Commission
IEEE	-	Institute of Electrical and Electronics Engineers
ILAC	-	International Laboratory Accreditation Cooperation
IP	-	Internet Protocol
ISO	-	International Organization for Standardization
IT	-	Information Technology
H/W	-	Hardware
HTTPS	-	Hypertext Transfer Protocol Secure
MPLS	-	Multiprotocol Label Switching
MU	-	Measurement Uncertainty
NABL	-	National Accreditation Board for Testing and Calibration Laboratories
NIST	-	National Institute of Standards and Technology
OWASP	-	Open Web Application Security Project
PAAS	-	Platform as a service
RFP	-	Request for Proposal
RUSP	-	Requirements for quality of Ready to Use Software Product
S/W	-	Software
SOP	-	Standard Operating Procedure
SQuaRE	-	Systems and Software Quality Requirements and Evaluation
SRS	-	Software Requirements Selection
SUT	-	Software under Testing
TM	-	Testing Methodology
UPS	-	Uninterrupted Power Supply
VAPT	-	Vulnerability Assessment and Penetration Testing
VPN	-	Virtual Private Network
WCAG	-	Web Content Accessibility Guidelines

1. INTRODUCTION

1.1 NABL has taken a proactive approach towards accreditation for conformity assessment of secure and intelligent digital systems, covering evaluation of functional performance, safety, cybersecurity and regulatory compliance of software-driven systems and intelligent products in various domains (medical, defense, etc.) and emerging technologies such as AI/ML, IoT, blockchain, embedded software technology with remote management, and SoC (System on Chip) as per ISO/IEC 17025:2017 "General requirements for the competence of testing and calibration laboratories." . This includes Static and Dynamic analysis or security vulnerabilities, functional and performance validation, safety assurance, and hardware-software integration testing. **Software testing is recognized as a crucial part of software validation, and this document emphasizes the need for robust testing practices as part of overall software verification and validation.**

1.2 The requirements in this specific criteria document are based on ISO/IEC 17025:2017 "General requirements for the competence of testing and calibration laboratories." It specifies competency and quality requirements particular to Software & IT System Testing Laboratories. This criteria document must be used in conjunction with ISO/IEC 17025:2017, providing an interpretation of that standard with sector-specific requirements. Further, the laboratory shall follow applicable international, national, regional, and local laws and regulations.

2. PURPOSE

2.1 Purpose of this document is to specify the requirements in addition and supplementary to ISO/IEC 17025: 2017 to assess the Software and IT System testing laboratories ensuring quality of Software under testing (SUTs) and IT systems. **It underscores that thorough testing is essential for validating software, and highlights the need for proper test method verification and validation (per Cl. No. 7.2 of ISO/IEC 17025:2017) to ensure test results are reliable.**

3. SCOPE

3.1 This document is applicable for assessing the competence of the laboratory performing the software and IT system testing for test parameters viz. Functionality, Performance/ Efficiency, Usability, Accessibility, Security Application Security, Network Security, Endpoint Security, Information Security, Cloud Security, etc), Reliability, Interoperability, Maintainability, Portability, Code review, Conformance (including Unicode) etc.

The Software and IT System may include but not limited to the following:

a) Telecom software/Protocol

b) Embedded systems.

c) Mobile Device and Mobile Applications

d) e-Governance Application and solution Evaluation

e) Data Analytics Software

f) Medical software and systems (including Software as a Medical Device).

g) Artificial Intelligence (AI/ML) and Large Language Models (LLMs) (including edge AI deployments)

h) Blockchain-enabled systems.

i) Cryptographic security (including readiness for post-quantum cryptography as per emerging NIST's FIPS standards)

j) Cyber-physical systems (CPS)

k) E-Procurement System Software

l) Process and control software

m) Web Application and Website

n) Linguistics Software (if there is linguistic implementation)

o) Sector specific Software & IT system e.g. Defence/ Railways/Banks/ Public sector etc.

p) Software product test & evaluation as per regulatory framework/ guidelines

q) Identification and Tracking Systems (RFID systems, Barcodes), Surveillance and Monitoring Systems (Video analytics and intelligent surveillance systems), Communication and Wireless Systems including Cellular, Satellite and LPWAN technology, Secure Hardware and Embedded Security (Secure elements SIM cards) , industrial automation and control systems components(IACS), Automotive and vehicular System Internet of Things and smart devices and respective connectivity platform gateways, firewalls, Networking and Data Communication Protocols, Cybersecurity and Data Protection Systems, Embedded Systems and Electronics, Sensor and Data Acquisition Systems (as per applicable standards and Essential Requirements).

3.2 Software/System Quality Parameters: The laboratory shall demonstrate competence in evaluating the following quality parameters, as applicable to the product or system under test:

1. Functionality – Verification of all functional requirements and use cases.
2. Performance / Efficiency – Load, stress, and performance benchmarking.
3. Usability – User experience and human factors evaluation.
4. Accessibility – Conformance to accessibility standards (e.g., WCAG).
5. Website testing – Website functional, compatibility, and compliance testing.
6. Security testing covering Security by design principles including Vulnerability Assessment, Penetration testing Application Security, Network Security, Endpoint Security, API security testing Information Security, computer system security, Cloud Security, etc
7. Crypto Agility & Cryptographic security (including readiness for post-quantum cryptography as per emerging NIST's FIPS standards)
8. Reliability – Error handling, fault tolerance, uptime, and consistency under expected operation.
9. Interoperability – Compatibility with other systems, data formats, and interfaces.
10. Maintainability – Code quality, modularity, and ease of maintenance.
11. Portability – Ability to operate in different environments or platforms.
12. Code review and secure coding compliance – Adherence to secure coding standards and best practices.
13. Security Assessment of software tools used for software development to insure not to generate malware or unused code.
14. Conformance to standards – Compliance with relevant industry-specific functional safety, regulatory standards, and emerging software/AI standards (e.g., Unicode, medical device software standards, aviation software standards, etc.).
15. Assessment of modern software-intensive and intelligent systems – Specialized evaluation of AI/ML models, IoT ecosystems, blockchain logic, etc., for trustworthiness and correctness.

4. TERMS AND DEFINITIONS

For the purpose of this document, the definitions and concepts given in ISO/IEC/IEEE 29119-1 shall apply.

 # Note 1: *For the purpose of this document, Software under Testing (SUT), wherever mentioned, shall mean Software and IT System.*

 # Note 2: *The Laboratory/ Institute/ Organization/ Service provider, which possesses the minimum required infrastructure and capability to perform software testing is termed as Software & IT System Testing Laboratory in this document.*

4.1 Software under Testing (SUT): Software, IT system, or intelligent product (including embedded system and electronics or AI-based systems) being evaluated by a testing laboratory for compliance to specified functional, security, performance, safety, or regulatory requirements.

4.2 Software & IT System Testing Laboratory: An organization or entity that performs testing and validation of software systems in accordance with ISO/IEC 17025:2017 and supplementary criteria specified in this document. It may cover testing in domains such as medical, defense, AI, IoT, information technology and cybersecurity.

4.3 AI System: An engineered system i.e. AI agent that generates outputs such as predictions, recommendations, or decisions influencing real or virtual environments, for a defined objective (as per ISO/IEC 22989).

4.4 Large Language Model (LLM): A category of generative AI models trained on extensive language datasets, designed to generate context- aware, human-like language output, used in a variety of intelligent systems.

4.5 Embedded System: A computer system with a dedicated function within a larger mechanical or electrical system, often with real-time computing constraints, integrating both hardware and software components.

4.6 Conformity Assessment: A demonstration that specified requirements relating to a product, process, system, or body are fulfilled (as per ISO/IEC 17000).

4.7 Code Review: A systematic examination of source code intended to identify bugs, security vulnerabilities, performance issues, and compliance with coding standards and best practices.

4.8 Cyber-Physical System (CPS): An integration of computation, networking, and physical processes where embedded computers monitor and control physical processes, usually with feedback loops.

4.9 Vulnerability Assessment and Penetration Testing (VAPT): Security testing methodology involving systematic scanning for vulnerabilities and simulated attacks to identify and evaluate system security weaknesses.

4.10 Blockchain-Based System: A digital system or software product leveraging distributed ledger technology (DLT) for immutable, transparent, and secure recording of transactions or interactions.

4.11 Accessibility: Accessibility refers to the extent to which software products, systems, or services can be used by people with the widest range of abilities and disabilities in a variety of contexts and environments. It ensures that users, including those with visual, auditory, motor, or cognitive impairments, can perceive, understand, navigate, and interact with the software effectively.

4.12 Interoperability: The ability of software or systems to exchange and use information seamlessly across different platforms or organizational boundaries.

4.13 Maintainability: The ease with which a software system can be modified to correct faults, improve performance, or adapt to a changed environment.

4.14 Portability: The ability of a software product to be transferred from one hardware or software environment to another with minimal adaptation effort.

4.15 Secure Coding: The practice of writing software that guards against vulnerabilities by following

security best practices and validated patterns to prevent exploitability.

4.16 Post-Quantum Cryptography (PQC): Cryptographic methods designed to secure data against threats from quantum computing, as being developed under FIPS-203 by NIST.

4.17 SaMD (Software as a Medical Device): Software intended to be used for medical purposes that performs these functions without being part of a hardware medical device.

4.18 OWASP Top 10: A globally recognized list of the top 10 critical security risks to web applications, published by the OWASP Foundation.

5. GENERAL REQUIREMENTS

5.1 Impartiality (Cl. No. 4.1 of ISO/IEC 17025: 2017)

a) The laboratory shall clearly define the responsibilities to avoid conflict of interest when the laboratory staff is holding additional responsibilities such as design, development, implementation, operation, maintenance etc. other than testing.

b) When the test scripts and tools are developed as part of the software development process, the laboratory shall have procedures for ensuring its independence and that of its staff from the development process and shall validate its suitability before use to avoid any potential conflict of interest.

5.2 Confidentiality (Cl. No. 4.2 of ISO/IEC 17025: 2017)

Operational production data containing personally identifiable information or any other confidential information shall be avoided for the purpose of carrying out testing. If personally identifiable information or otherwise confidential information is used as test data, it shall be justifiable, masked and protected either by removal or modification.

Refer ISO/IEC 27002: 2022 Clause 8.12 (Data leakage prevention) for guidelines provided under protection of test data (specifically clause 14.3 in the 2005 edition).

6. Structural Requirements

Laboratory shall comply with the requirements as per Cl. No. 5 of ISO/IEC 17025:2017 and relevant NABL policies prevalent at that time.

7. Resource Requirements

7.1 General (Cl. No. 6.1 of ISO/IEC 17025: 2017)

a) The availability of the resources (personnel, environment, facility, equipment) shall be ensured even while carrying out at remote location at Data Centre, cloud and customer site facility or accessing the remote system from permanent facility through secured connectivity.

b) The laboratory should also ensure it has or can access specialized resources required for emerging technologies. This includes maintaining suitable testbeds, simulators, or tools for domains like IoT (e.g., sensor networks, device simulators), AI (e.g., model testing frameworks, datasets for validation), blockchain (e.g., smart contract testing platforms), etc. The availability and readiness of these resources should be confirmed as part of the lab's resource planning to avoid limiting the scope of testing.

7.2 Personnel (Cl. No. 6.2 of ISO/IEC 17025: 2017)

a) The laboratory shall document the minimum competence requirements for each function such as Test Manager/ Lead and Tester/ Evaluator/ Lead Evaluator etc.

The competency requirements include:

i. Relevant Training, Qualification and Education,

ii. Domain knowledge of the software and IT System being tested,

iii. Knowledge of relevant standard,

iv. Skill of the personal involved in the evaluation/Testing methodology, and

v. Required experience for all the functions.

b) Lab shall identify the appropriate personnel for report, review and authorization to release the test report to the customers.

7.3 Facilities and environmental conditions (Cl. No. 6.3 of ISO/IEC 17025: 2017)

a) "Test environment" shall comprise of hardware, reference sample as S/W, H/W standard reference, (e.g. standard finger, standard patterns etc.), platforms, middle wares, network, and associated software libraries, drivers etc. on which the software being tested is running.

b) Software Testing service may be carried out only in the appropriate test environment that may be created & accessed by following means-

i. **Permanent/ in house testing service:** Test is executed by creating test environment within the laboratory.

ii. **On site Testing:** Test is executed on site location i.e. away from permanent laboratory, by creating test environment accessing IT system infrastructure at customer location or IT infrastructure at Data centres.

iii. **Accessing 'on site testing facility' at permanent laboratory:** Test is executed at permanent laboratory by remotely accessing Test environment that is created either at 'on site customer location' or Data centers or on cloud using IAAS/PAAS services.

In each of the above cases, the laboratory shall maintain and record the configuration of test environment set up throughout the testing process. Measures to control shall be defined and implemented wherever required.

c) In case of In-house testing, where the testing is required to be performed using IT infrastructure or tools or system controlled by the customer/ developer, procedures for the extent of control on these items shall be defined and recorded.

d) Testing shall be performed in a realistic/ similar to production/ target environment relevant to the test parameter and context of use. Deviation, if any, shall be justifiable and the risks due to test environment related specific failures shall be recorded.

e) Testing should be conducted in an environment segregated from both the production and development environment.

f) There should be no other concurrent activities or interference from other activities occurring during testing that may affect or invalidate the results.

g) Where any virtual environment or other special configuration (creating virtual users, setting network bandwidth etc. in case of performance/ load testing) is created for simulating real life production conditions, it shall be fully documented in the test records along with a justification as to why it is believed not to affect or invalidate the results.

h) Access control to laboratory areas, server room, Data center and designated test areas (e.g. Usability Laboratory, Common Criteria laboratory etc.) shall be restricted to ensure the integrity of SUTs, test environment and test artefacts.

i) Risk management process, electronic delivery process, security controls, secure transmission and handling procedures.

7.4 Equipment (Cl. No. of 6.4 of ISO/IEC 17025: 2017)

a) Software Test tools along with its hosting platform and other associated plug-in & add-ons and associated devices/standard required for test are considered as equipment. The laboratory shall have access to equipment and same shall be brought under configuration management.

b) The laboratory shall ensure that current version of test tools and configuration is maintained with regular updation of patches.

c) When the laboratory uses equipment outside its permanent control such as usage of Test tools provided by customer, availing software testing tools on cloud, use of open-source tools etc, it shall ensure and maintain records at the time of use but not limited to the following:

   i. Validations of open-source tools. Open-source components are managed as defined by the best practices.

   ii. the access control during the test is demonstrable by a document like agreement etc.

   iii. The license and version details including patch particulars,

   iv. Configuration management and settings details at the time of use,

   v. Repository of automated Test Suite and test data used,

   vi. Identity of users and role bases access details,

   vii. Type of connectivity Viz. HTTPS, VPN/ IPSec, MPLS/ leased line etc and bandwidth details,

   viii. Control of test data (requirement that Software test tools should be reset or logs emptied between tests to ensure that only current test data is recorded)

d) The laboratory shall verify that Software Test tool meets the intended requirements and maintain the records of installation but not limited to the following:

   i. Version Number and patch details along with license key details.

   ii. Records of verification carried out along with required configuration settings.

   iii. Identification of user roles and Access/privilege rights

   iv. Configuration details of installation of each instance with unique identification.

e) The laboratory shall maintain records of equipment and shall at least address the following:

   i. Identity – each instance of software/hardware.

   ii. Supplier/Developer name and version number.

   iii. Checks - installation/operational qualifications

iv. Location – target system name or location.

v. Manufacturer's instructions – user manuals.

vi. Validation details.

vii. Up gradation and renewal of license

7.5 Externally provided products and services (Cl. No. 6.6 of ISO/IEC 17025: 2017)

7.5.1 Externally provided products and services in software testing includes:

a) Products:

i. Commercially available software testing tools/ test suites, Configuration Management tools, Test Management Tools, Defect management tools etc.

ii. IT infrastructures such as gateways, Firewalls, Connectivity Management Platform, use case Servers, Secure Storage devices, General storage devices, Client machines (End nodes enabled with computation or non-computational), Network components, etc.

iii. System software, APIs and Middleware

iv. Protection of Supply chain data through an appropriate set of security controls.

b) Services:

i. Testing tools, IAAS, PAAS services etc. on cloud

ii. Internet/leased line connectivity services

iii. Remote management for Secure firmware update and patch installation

iv. Hardware and software/tool maintenance/ upgradation services

v. Network maintenance services

vi. Maintenance of UPS, air conditioners etc.

7.5.2 While availing services such as Cloud services, Internet services, maintenance services, laboratory shall establish requirements in terms of Service Level Agreement and monitor the performance whether desired level of service is delivered.

8. PROCESS REQUIREMENTS

8.1 Review of requests, tenders and contracts (Cl. No. 7.1 of ISO/IEC 17025: 2017)

8.1.1 The procedure for the review of requests, tenders and contracts at least shall address details of Software under Test (SUT): Product Title, Version number, Release details, brief description etc.

a) Test environment and interface requirements including establishing test environment at customer premises, cloud, data center etc.

b) SUT identification: Clear identification of the Software Under Test – including the product/software name, version number, release identifiers, and a brief description of the SUT's function or purpose.

c) Test parameters: The types of testing to be performed, classified broadly into functional and non-functional categories. This should list which aspects are in scope, such as functionality, performance, security, usability, interoperability, etc., as applicable to the SUT. (For example: "Functional testing, performance (load) testing, and security (OWASP top 10) testing including security by design principle, will be performed.")

d) Reporting Methodology.

8.1.2 SRS (Software Requirement Specification), Functional/ Non-functional requirement specification (FRS), Use case document, Business process document, RFP document, Design/ Architecture/interface related documents etc. as relevant against which functional/non-functional testing is to be performed.

8.1.3 The laboratory shall identify the test methods, standards, or specifications that will be used for testing the SUT. This includes international or national standards (e.g., Common Criteria ISO/IEC 15408 for security evaluation, GIGW guidelines for website quality, OWASP guidelines for application security, etc.), industry best-practice guides, and any specific test tools or test data requirements. For example, the review should note if testing will follow the OWASP Web Security Testing Guide, use certain automated scanning tools, or require specific test data sets. It should also capture any middleware, supporting software, or test harness needed.

8.1.4 Any requirements for retesting or regression testing should be defined at the review stage. For instance, if the contract involves the possibility that the software will be re-tested after fixes (bug fixes or updates) within the project scope, the conditions for that (how many cycles of re-test are included, turnaround time, etc.) should be agreed upon.

8.1.5 Domain or technology-specific training or competency requirements should be identified. If the SUT involves a specialized domain (for example, an aerospace control system or a blockchain application), the lab should confirm whether its personnel need any briefing or training on domain specifics or if subject matter experts need to be engaged. This ensures that the team is adequately prepared to understand the context of the SUT.

8.1.6 When the laboratory uses equipment such as Test tools, computer systems, platforms, etc. provided by customer, laboratory shall review the applicability and ensure its verification before use with license and version details including patch particulars

8.1.7 Acceptance criteria for the SUT testing should be defined where applicable. This refers to the criteria by which the test results will be judged acceptable or not. For example, for performance testing: response time must be under X seconds under Y load; for compliance testing: the software passes all required checklist items; etc. If the client has defined success criteria (like "no Critical or High severity defects open" or "all user requirements tested and passed"), these should be recorded

8.1.8 The laboratory's review process shall also include identification of any critical functional areas and potential security risks of the SUT that require special attention during testing. This means the lab, based on the information available, will pinpoint key features that are safety-critical or business-critical, and any known threat scenarios or sensitive data flows in the SUT. By

recognizing these crucial points up front (for instance, an authentication module in a banking app, or a machine learning decision component in an AI system), the lab can plan to prioritize and thoroughly test these areas. This practice helps ensure that testing efforts are focused on areas of highest risk or importance, aligning with a risk-based testing approach right from the planning stage.

8.2 Selection, verification and validation of methods (Cl. No. 7.2 of ISO/IEC 17025: 2017)

8.2.1 Since most of the Software and IT system testing are tool dependent and IT standards are guideline in nature; laboratory shall develop test methods / procedures which shall be of following categories:

- In case of test methods based on standard commercial tools, laboratory shall document Test procedure/ SOP's detailing the application of the tool, its scope, test implementation such as test pre-conditions with test environment creation, test execution and logging of results including removal of false positives. However, Laboratory shall verify that the documented test procedure/SOP's meets the performance specification before put into use.
- In case of test methods based on open-source tools, community developed tools or laboratory developed tools, laboratory shall document test method detailing the application of the tool, its scope, test implementation such as test pre-conditions with test environment creation, test execution and logging of results including removal of false positives. However, laboratory shall validate the test method conforming its intended use before put into use. Some of the validation guidelines given in annexure may be followed to establish its intended use.

8.2.2 The verification/ validation records of Test methodologies/SOPs shall address the expected outcomes and deliverables as applicable as per Annex – C.

8.2.3 The selection and combination of Test Design techniques/ Test methods shall ensure that:

- Test case results are not ambiguous and have single thread of execution with objective results relating to expected outcomes.
- The traceability between the test basis, feature sets, test conditions, test coverage items, test cases, test sets and test procedures (and/or automated test scripts) exists and is recorded.
- The laboratory shall periodically review and approve the test artefacts such as Test plan, Test design specification, Test cases, Test procedures (and/or automated test scripts), Test logs and test incident report before execution or release.
- All test artefacts shall be brought under configuration management.

8.3 Sampling (Cl. No. 7.3 of ISO/IEC 17025: 2017)

8.3.1 The sampling is not for selection of product for testing. The sampling in IT activities shall be done to optimize the Test activities.

8.3.2 Test strategy/ approach shall determine the extent of sampling in terms of test analysis and activities to be performed based on the context and risk of testing. The laboratory shall establish and confirm the extent of sampling by creating the test artefacts as relevant but not limited to the following:

- Test Plan specifying the risk, strategy/ approach of testing, test techniques to be used, exit criteria, etc.
- Test design specification specifying either the features/functions/transaction/quality attribute or structural element to be tested and their corresponding test conditions.
- Test case specification specifying selection and prioritization of test cases, etc
- Extent and Selection of regression tests to rerun;
- Selection of source code to review based on risk.

8.4 Handling of test items (Cl. No. 7.4 of ISO/IEC 17025: 2017)

a) SUT shall be clearly identified with its version number along with the release note of SUT as applicable, from the customer.

b) Any subsequent testing after defects are fixed, shall be carried out only after SUT is submitted with clear identification of different version.

c) In case of installation problems such as compatibility with middleware, libraries, etc. and/or integration with other systems, system software etc., the laboratory shall record the same and customer shall be accordingly intimated for further instructions before proceeding.

d) Maintain & record the configuration management with appropriate metadata to ensure it is unique. Test base documents pertaining to SUT such as SRS, FRS, use case document, RFP, design/ architecture/ interface documents etc shall also be brought under configuration management.

e) The laboratory shall ensure integrity of the SUT and shall establish method to verify throughout the test life cycle. It shall be protected from unauthorized/ unintended modification or changes during normal installation process, integration with other systems/tools, testing process etc.

f) In case an unintended modification occurs in the test item software or gets corrupted, the software shall be re-installed to its initial state before resuming the test.

g) Access to the Test environment and SUT shall be controlled.

h) Laboratory shall ensure that the integrity of SUT and test environment is maintained throughout test life cycle when installed outside laboratory's control such as in Data centre, cloud, customer premises etc.

8.5 Technical records (Cl. No. 7.5 of ISO/IEC 17025: 2017)

a) Technical records include all test deliverables/ artefacts as listed in **annexure A** which are produced during testing besides Test environment set up details, equipment/ tools details, SUT and its allied documents, contract review records etc.

b) All technical records shall be brought under configuration management. Access to technical records shall be controlled and privileged rights shall be defined.

c) privileged rights shall be defined.

8.6 Ensuring the validity of results (Cl. No. 7.7 ISO/IEC 17025: 2017)

The procedure for monitoring the validity of results shall address the following and records shall be maintained as practicable:

i. Periodic review of Test plan covering Test strategy, Risk, scope, schedule and resources.

ii. Periodic review of Test design specification and Test case design records verifying for appropriateness of Test condition/scenario selection, Test method selection, test coverage with back and forth traceability, Test environment setup, Test tool selection, granularity of testing etc.

iii. Review of Test execution logs and incident reporting.

iv. Review of false positives in the report generated by automated tools.

v. Functional check(s) and intermediate checks of Test tools.

vi. Review of regression Test suites.

8.7 Reporting of results (Cl. No. 7.8 of IS/IEC 17025: 2017)

In addition to the requirement of Clause 7.8 of ISO/IEC 17025: 2017 following may be included wherever applicable:

i. In case of multiple reports issued for same SUT such as separate report for functionality,

security, performance, Usability, interoperability etc., the laboratory shall ensure linkage between the reports confirming the test requirements as per the contract review.

ii. Description of open defects in an unambiguous way

iii. Traceability of test results with requirements and/or specifications

iv. Information on specific test conditions and simulation criteria

v. Deployment (In-house, cloud, data centre etc.) and configuration details of test environmental set up including mode of access.

vi. Reference to Test methods including any deviations or exclusions

vii. Details of Test tools along with version and patch details and any configuration settings, if any

viii. Methodology used for Severity Classification for Software anomalies such as IEEE 1044, user requirement etc.

ix. Methodology used for Regression testing

8.8 Actions to Address Risks and Opportunities

a) Depending upon the context, constraints and type of testing, the laboratory shall adopt appropriate risk- based approach to testing. Risk identification and analysis methodology as relevant may be adopted so that the perceived risks in the developed/delivered system are identified, scored, prioritized and subsequently mitigated.

b) The laboratory shall identify and address risk in test plan by specifying appropriate risk mitigation strategy and approach to testing.

c) The laboratory shall evaluate the effectiveness of risk implementation by implementing actions such as review of test plan, review of test coverage, evaluation of test exit criteria etc.

d) **For laboratories engaged in testing systems based on AI/LLM, blockchain, or IoT technologies, the organization shall incorporate controls to mitigate risks related to bias, algorithmic integrity, data privacy, and embedded system security.** This may include independent review of AI model results, oversight for ethical considerations, and separation of duties to prevent conflicts of interest.

Note: In context of software testing, the risks may exist/ arise at any stage/ action e.g. risks of not satisfying regulatory and/or legal requirements, failing to meet contractual obligations, criticality of the product, non-availability/ scarcity of required infrastructure, lacking simulation of real testing environment, delivery schedule,

Annexure -A
Technical Records

All technical records created, received, or used during testing activities, including but not limited to the following:-

1. Test Deliverables / Artefacts
    - Test Plan
    - Test Design Specification
    - Test Case Specification
    - Test Scripts / Test Data
    - Traceability Matrix
    - Test Execution Logs
    - Test Summary Report
    - Defect Logs
    - Regression Test Logs
    - Risk Analysis Reports
    - Code Review Records
    - Compliance and Conformance Checklists
2. Test Environment Setup Details
    - OS version, APIs, middleware, runtime libraries, tools, database, cloud configurations, etc.
3. Equipment and Tools

    Details of tools used for:
    - Automated testing
    - Performance/load testing
    - VAPT/security tools
    - Configuration/version control tools
4. Software Under Test (SUT) and Allied Documents
    - SUT version and release notes
    - Installation and configuration guides
    - Release build metadata
    - Source code snapshots (if applicable)
    - Supporting technical documents such as:
        - SRS/FRS
        - Design/Architecture Documents
        - Interface Definitions
        - AI Model documentation, if applicable
5. Contract Review Records
    - Test request form
    - Scope of testing and acceptance criteria
    - Customer communications, requirement clarifications, and approval of deliverables

Annexure – B
Test Tool Validation and Traceability

1. The laboratory shall establish and maintain traceability of test results for all test tools (including open-source, third-party, and customer-supplied tools) used in the testing of software and IT systems. This is to ensure the validity, repeatability, and reproducibility of test results. All test tool software and any measurement or output they produce should be linked to known references or evidence of correctness, analogous to calibration in physical measurements.
2. Traceability shall be demonstrated through:
   a) Validation and verification of test tools – whether the tool is manual or automated, for performance testing, security testing, etc., the lab must verify that the tool works as intended. For example, if a performance benchmarking tool is used, the lab should have verified that it accurately measures response times under known conditions.

   b) Documented evidence of tool performance – there should be records showing that the tools perform as expected for their specific applications. This includes evidence for specialized domains: e.g., in medical or defense software testing, proof that tools have been checked against known standards or reference cases in those domains.

   c) Benchmarking against standards or references – where applicable, the laboratory should calibrate or benchmark the test tool's performance against standard reference data or known benchmarks. For instance, a security scanner might be tested against a reference set of vulnerabilities to ensure it detects them, or an AI model output validator might be checked against a known dataset.

   d) Use of reference artifacts: the lab should, when possible, verify and re-verify tools using stable reference artifacts (such as a "golden" version of a software or a dataset with expected outcomes) to ensure the tool yields consistent results over time.

3. When open-source tools or cloud-based testing services are used, the laboratory shall maintain specific records to ensure repeatability:
   a) Version details and patch levels of the tool at time of use (to ensure results can be replicated with the same tool version).

   b) Validation or verification reports for the tool, along with snapshots or documentation of critical configurations or settings applied during testing.

   c) Change control records for any updates or modifications made to the tools or test environments, including impact analysis on test outcomes if a tool is updated during an ongoing project.

4. For domains requiring higher assurance levels (such as medical or defence software testing), the laboratory shall take extra measures to ensure tool reliability and traceability
   a) Tools should be validated using certified datasets or reference samples whenever available (e.g., using an FDA-approved reference dataset for a medical imaging software test, or known military standard test vectors for defense systems).

   b) Documentation for such tools and tests should explicitly link test outcomes to standard protocols, or simulated environments that have a known baseline. Essentially, the lab's records should make it clear how one can trace a test result back to a reliable reference or method.

FOR EMERGING TECHNOLOGY

For emerging technology domains (AI, LLMs, blockchain, IoT, etc.), test tools must be validated in context-sensitive ways:

a) For AI and LLM testing tools – the datasets and scenarios used for validating the tool's correctness must be recorded, with dataset versions controlled and results reproducible. For example, an AI bias detection tool should be run on a benchmark dataset and produce expected bias metrics to confirm it works.

b) For blockchain testing tools – the laboratory should document aspects like transaction logs, hash validations, or smart contract execution environments used to validate the tool. These should be reproducible; for instance, a smart contract analysis tool might be tested on a known set of vulnerable contracts to ensure it catches expected issues.

c) For IoT testing tools – if using IoT device simulators, network emulators, or interoperability test beds, the lab must document their configurations fully. There should be full traceability from the test outcomes back to the test bed setup (e.g., if an IoT protocol analyzer is used, the version of the simulator firmware and network conditions should be recorded so results can be traced and reproduced).

d) In all cases, the laboratory shall ensure that test tools and platforms are uniquely identified and that their test environments can be recreated. This means the lab should be able to take its documentation and set up the same test tool environment (software versions, configurations, datasets, etc.) to reproduce a test if needed. If any deviations from standard procedures occur (for example, using a tool beyond its typical use or modifying a tool), these deviations must be justified validated, documented, and accompanied by a risk assessment and mitigation plan.

e) The laboratory shall have procedures to periodically review and re-validate all critical test tools and associated environments, updating them as necessary to remain compliant with evolving standards and emerging technology requirements. (For example, if new versions of a security standard are released or new vulnerabilities are discovered in a test tool, the lab should update its tools and validate again.)

Note: The lab should retain any necessary reference artifacts (datasets, old software versions, etc.) to support ongoing traceability and consistency of results over time.

Annexure – C
Test methodologies

S. No.	Test Process/ Methodologies	Test Deliverables/ Artifacts	Outcomes
1.	Test Planning Process	- Test Plan	- Scope of testing - Risk that can be treated by testing are identified, analyzed and classified - Test strategy, test environment, test tool and test data needs are identified; - Resource, training and staffing needs - Analysis, design, implementation, execution and reporting schedule - Exit criteria
2	Test Design and Implementation	- Test Design Specification - Test case design - Test Procedure Specification	- The test basis for each test item is analysed; - The features to be tested are combined into Feature Sets - The Test Conditions are derived - The Test Coverage Items are derived; - Test Cases are derived; - Test Sets are assembled; - Test Procedures are derived
3	Test Execution Process	- Test Execution log - Test Results	- Execute Test Procedure(s) - Compare Test Results - Record Test Execution
4	Test Incident Reporting Process	- Test Incident reporting	- Test results are analysed - Incidents are confirmed - Incident report created/updated - The status and details of previously-raised incidents are determined; - Risk matrix to be prepared w.r.t previously-raised incidents.

9. REFERENCES

- ISO/IEC/IEEE 29119: Software and systems engineering - Software testing
- ISO/IEC 25010: Systems and software engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and software quality models
- ISO/IEC 25051: Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) - Requirements for quality of Ready to Use Software Product (RUSP) and instructions for testing.
- ISO/IEC 27002: For guidelines provided under protection of test data (Cl no14.3).
- ISO/IEC 16085: Systems and Software Engineering - Life Cycle Processes - Risk Management
- IEEE 1044-2009 - IEEE Standard Classification for Software.
- ISO/IEC 20243 - Information Technology – O-TTPS – Mitigating maliciously tainted and counterfeit products

National Accreditation Board for Testing and Calibration Laboratories (NABL)
J200, World Trade Centre,
Nauroji Nagar, New Delhi-110029
Website: www.nabl-india.org
Tel. No. - +91-11-40032400 (30 lines)