Designation: E 2066 – 00

Standard Guide for Validation of Laboratory Information Management Systems[1]

This standard is issued under the fixed designation E 2066; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

1. Scope

1.1 This guide describes an approach to the validation process for a Laboratory Information Management System (LIMS).

1.2 This guide is for validation of a commercial LIMS purchased from a vendor. The procedures may apply to other types of systems, but this guide makes no claim to address all issues for other types of systems. Further, in-house developed LIMS, that is, those developed by internal or external programmers specifically for an organization, can utilize this guide. It should be noted that there are a number of related software development issues that this guide does not address. Users who embark on developing a LIMS either internally or with external programmers also should consult the appropriate ASTM, ISO, and IEEE software development standards.

1.3 This guide is intended to educate individuals on LIMS validation, to provide standard terminology useful in discussions with independent validation consultants, and to provide guidance for development of validation plans, test plans, required standard operating procedures, and the final validation report.

2. Referenced Documents

2.1 *ASTM Standards:*

E 622 Guide for Developing Computerized Systems[2]
E 623 Guide for Developing Functional Requirements for Computerized Systems[3]
E 624 Guide for Developing Implementation Designs for Computerized Systems[3]
E 627 Guide for Documenting Computerized Systems[2]
E 919 Specification for Software Documentation for a Computerized System[2]
E 1013 Terminology Relating to Computerized Systems[2]
E 1384 Guide for Content and Structure of the Electronic Health Record (EHR)[2]
E 1578 Guide for Laboratory Information Management Systems (LIMS)[2]
E 1639 Guide for Functional Requirements of Clinical Laboratory Information Management Systems (CLIMS)

2.2 *IEEE Standards:*[4]

100 Standard Dictionary of Electric and Electronic Terms
610 Standard Glossaries of Computer-Related Terminology
729 Glossary of Software Engineering Terminology
730.1 Standard for Software Quality Assurance Plans
730.2 Guide for Software Quality Assurance Plans
828 Standard for Software Configuration Management Plans
829 Standard for Software Testing Documentation
830 Guide for Software Test Documentation
1008 Standard for Software Unit Testing
1012 Standard for Software Verification and Validation Plans
1016 Recommended Practice for Software Design Descriptions
1028 Standard for Software Reviews and Audits
1042 Guide to Software Configuration Management
1058-1 Standard for Software Project Management Plans
1063 Standard for Software User Documentation
1074 Standard for Developing Software Life Cycle Processes
1228 Standard for Software Safety Plans

2.3 *ISO Standards:*[5]

9000 Quality Management and Quality Assurance Standards - Guidelines for Selection and Use
9000-3 Guidelines for Application of ISO 9001 to Development, Supply, and Maintenance of Software
9001 Quality Systems—Model for Quality Assurance in Design, Production, Installation, and Servicing
9002 Quality Systems—Model for Quality Assurance in Production and Installation
9003 Quality Systems—Model for Quality Assurance in Final Inspection and Test
9004 Quality Management and Quality System Elements—Guidelines
9004-2 Quality Management and Quality System Elements, Part 2 Guidelines for Services
9004-4 Guidelines for Quality Improvements

[1] This guide is under the jurisdiction of ASTM Committee E13 on Molecular Spectroscopy and Chromatography and is the direct responsibility of Subcommittee E13.15 on Analytical Data.

Current edition approved Jan. 10, 2000. Published March 2000.

[2] *Annual Book of ASTM Standards*, Vol 14.01.

[3] Discontinued 1994; see *1993 Annual Book of ASTM Standards*, Vol 14.01.

[4] Available from Institute of Electrical and Electronic Engineers, Inc., 445 Hoes Lane, P. O. Box 1331, Piscataway, NJ 08855–1331.

[5] Available from International Organization for Standardization, 1 rue de Varembé, Case postale 56, CH-1211 Genevé 20, Switzerland.

10005 Guidelines for Quality Plans
10007 Guidelines for Configuration Management
10011-1 Guidelines for Auditing Quality Systems, Part 1 Auditing
10011-2 Guidelines for Auditing Quality Systems, Part 2 Qualification Criteria for Auditors
10011-3 Guidelines for Auditing Quality Systems, Part 3 Managing Audit Programs
8402 Quality Vocabulary
2382 Data Processing Vocabulary

3. Terminology

3.1 *Definitions*—This guide defines terminology used in the validation of computerized systems. The standards listed in Section 2 provide additional definitions that the reader may want to review before beginning their validation process.

3.1.1 *acceptance criteria*, *n*—the specifications used to accept or reject a computer system, application, function, or test action.

3.1.2 *change control*, *n*—the process, authorities for, and procedures to be used to manage changes made to a computerized system or a system's data, or both. Change control is a vital activity of the Quality Assurance (QA) program within an establishment and should be described clearly in the establishment's SOPs.

3.1.3 *configuration management*, *n*—a discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a configured item, to control changes to those characteristics, to record and report change implementation status, and to verify compliance with specified requirements. **IEEE**

3.1.4 *customization*, *n*—the process of adding new software to or altering a LIMS so that it may perform functions not planned by the original system designers. This entails creating new software, compiling software modules, and linking modules to produce new executable programs. If done by the vendor, it may be considered and validated as part of the vendor system. See related definition for "customized system" in Terminology E 1013.

3.1.5 *delivered system*, *n*—the LIMS, as initially supplied by the vendor before any static configuration data have been added. In some cases, the vendor may contract with the laboratory to enter some configuration data on behalf of the laboratory, in which case the delivered system is still considered to be the default system before such customer-specific information has been added. When the vendor performs this task, they are an agent of the laboratory, and the customer shall meet the on-site validation requirements in Section 7.

3.1.6 *dynamic testing*, *n*—the actual testing of various functions and procedures using the LIMS software while in operation.

3.1.7 *installation qualification (IQ)*, *n*—documented verification that all key aspects of the installation adhere to approved design intentions as defined in system specifications and that manufacturers' recommendations are suitably considered.

3.1.8 *LIMS*, *n*—acronym for Laboratory Information Management System that refers to computer software and hardware that can acquire, analyze, report, store, manage data, and process information in the laboratory.

3.1.9 *LIMS data loading (configuration)*, *n*—the process of entering static data into appropriate data structures, such as tables or database records, to make a LIMS suitable for operation in a particular laboratory. This information may include items like names and addresses of laboratory customers, names of laboratory personnel, descriptions of tests performed by the laboratory, specifications, calculations, templates, or descriptions of LIMS reports, etc. In this process, no new functionality is added to the LIMS that was not originally planned by the system designers. Addition of configuration data may affect the behavior of the system.

3.1.10 *LIMS tailoring*, *n*—see *LIMS data loading (configuration)*.

3.1.11 *operational qualification (OQ)*, *n*—documented verification that each unit or the entire system operates as intended throughout its full operating range.

3.1.12 *quality assurance unit (QAU)*, *n*—the body of individuals responsible for design and interpretation of quality standards, such as validation procedures and processes (not product testing).

3.1.13 *source code*, *n*—a computer program expressed in human-readable form (programming language) that shall be translated into machine-readable form (object code) before it can be executed by the computer.

3.1.14 *static testing*, *n*—a structured review of the source code.

3.1.15 *stress testing*, *n*—the running of test protocols designed to test the limits of LIMS functions.

3.1.16 *test plan*, *n*—see *test protocol*.

3.1.17 *test protocol*, *n*—a written procedure describing a set of actions and their expected outcomes that when executed provides documentary evidence that specific functional requirements for the LIMS work as specified.

3.1.18 *validation*, *n*—the process of establishing documented evidence that provides a high degree of assurance that a specific process, system, or item consistently meets its predetermined specifications or quality attributes.

3.1.19 *validation plan*, *n*—the document that identifies all systems and subsystems involved in a specific validation effort and the approach by which they will be qualified and validated, including identification of responsibilities and expectations.

3.1.20 *validation team*, *n*—the group of individuals responsible for the validation process. This team may consist of representatives of the laboratory, QAU, Management Information System (MIS) organizations, or outside consultants.

3.1.21 *vendor audit*, *n*—an independent review and examination of system records and activities in order to test the adequacy and effectiveness of data security and data integrity procedures, to ensure compliance with established policy and operational procedures, and to recommend any necessary changes. **ANSI**

3.1.22 *vendor audit team*, *n*—a team made up of individuals who are knowledgeable in computer system engineering, auditing practices, computer system quality methods, regulatory compliance, validation practices, business and legal policies and procedures (applicable only to computer hardware and

software procurement and related services). (1)[6]

3.1.23 *version control*, *n*—control of all associated software and document versions. This also includes all documents associated with implementation, validation, or operation of a LIMS.

4. Significance and Use

4.1 Validation is an important and mandatory activity for laboratories that fall under regulatory agency review. Such laboratories produce data upon which the government depends to enforce laws and make decisions in the public interest. Examples include data to support approval of new drugs, prove marketed drugs meet specifications, enforce environmental laws, and develop forensic evidence for trial. This also extends to LIMS used in environmental laboratories. In some cases these systems may need to be interoperable with CLIMS and computer-based patient records (CPR) for reporting environmental exposures and clinical laboratory testing for biologic measure of stressor exposure. The enormous financial, legal, and social impact of these decisions requires government and public confidence in laboratory data. To ensure this confidence, government agencies regularly review laboratories operating under their rules to confirm that they are producing valid data. Computer system validation is a part of this review. This guide is designed to aid users validating LIMS and incorporating the validation process into their LIMS life cycle.

4.2 Validation must provide evidence of testing, training, audit and review, management responsibility, design control, and document control, both during the development of the system and its operation life (2).

5. The LIMS Life Cycle and the Validation Process

5.1 The process of validation should start at the beginning of the LIMS life cycle as defined in Guide E 1578. Adding validation to the end of the LIMS implementation could add from three to twelve months to the LIMS project. Further, adding validation to the end of the process would prevent the organization from using the LIMS during validation. Fig. 1 represents points where validation may impact the procurement of LIMS. Validation will not have an impact on all of the LIMS life cycle, and the amount of interaction with the validation team will vary during each life cycle phase.

5.1.1 *Validation Team Formation Phase*—This phase is typically not a separate phase in the LIMS life cycle, however, it is a critical part of the validation process. A typical team consists of representatives from the laboratory, MIS group, and QAU. There may be other team members depending on the scope of the project and resources within the organization. If required, the identified validation team members should begin to identify training courses on computer systems validation at this time. No training should take place until those who have been selected for the validation team have their management's full agreement to participate in this activity. These courses can be either in-house or outside-developed courses. The vendor audit team may consist only of the validation team or it may be a specific subgroup within the organization. It is recommended that the vendor audit team should include organizational members from the QAU, MIS, and the laboratory (1).

5.2 *Business Requirements Definition Phase*—The business unit, specifically the laboratory, shall contact the QAU to determine current good manufacturing practices (cGMPs), good manufacturing practices (GMPs), good automated laboratory practices (GALPs), and other requirements that shall be addressed with this project. An initial selection of validation team members is made at this time.

5.3 *Project Definition Phase*—Final agreement and management acceptance for all validation team members should be obtained. Because validation is complex and can take a long time, each team member should have the full support of their management. It is critical that management understands and agrees to the time commitment for these individuals. Without agreement from each member's management chain, the probability for developing and validating the LIMS successfully will diminish. Once formed, the validation team can start to address high-level issues such as the existence of corporate standard operation procedures (SOPs) needed for validation. Time constraints and inexperience of team members can be a limiting factor in the validation process. This is when the team should identify outside consultants that may be needed in the validation process and begin developing the validation plan. Appropriate training of validation team members also should be carried out during this phase of the LIMS life cycle.

5.4 *Model of Current State of Laboratory Practice*—The validation team typically is not part of this process.

5.5 *Model of Future State of Laboratory Practices*—The validation team typically is not part of this process.

5.6 *Functional Requirements Development Phase*—The validation team should work with the group responsible for developing functional requirements. At this time the team can also begin to develop and revise, as necessary, a high-level draft of the organization's validation plan for this project. The validation team may want to begin developing the high-level test protocols during this phase. Further this activity begins to focus attention on validation at the start of the project. Each identified functional requirement should be the subject of one or more test protocols.

5.7 *Request for Proposal (RFP) Phase*—The validation team shall ensure that the RFP includes both a request to audit the vendor and their validation requirements. People using this document for acceptance testing who are in unregulated industries may not require this audit process. Also, the validation team should request that the vendor's development process and LIMS application have undergone independent evaluation/validation. If another company, that is, a third party consultant or another corporation, has validated the vendor operation and LIMS development process, it does not mean that the prospective buyer can assume that the software is validated. During this time the team should specify what actions to take if a LIMS vendor denies them the right to an audit. The validation team should review the RFP prior to its submission to the vendor.

5.8 *Evaluation and Selection Phase*—The validation team should identify those people who will participate in vendor

[6] The boldface numbers in parentheses refer to the list of references at the end of this standard.

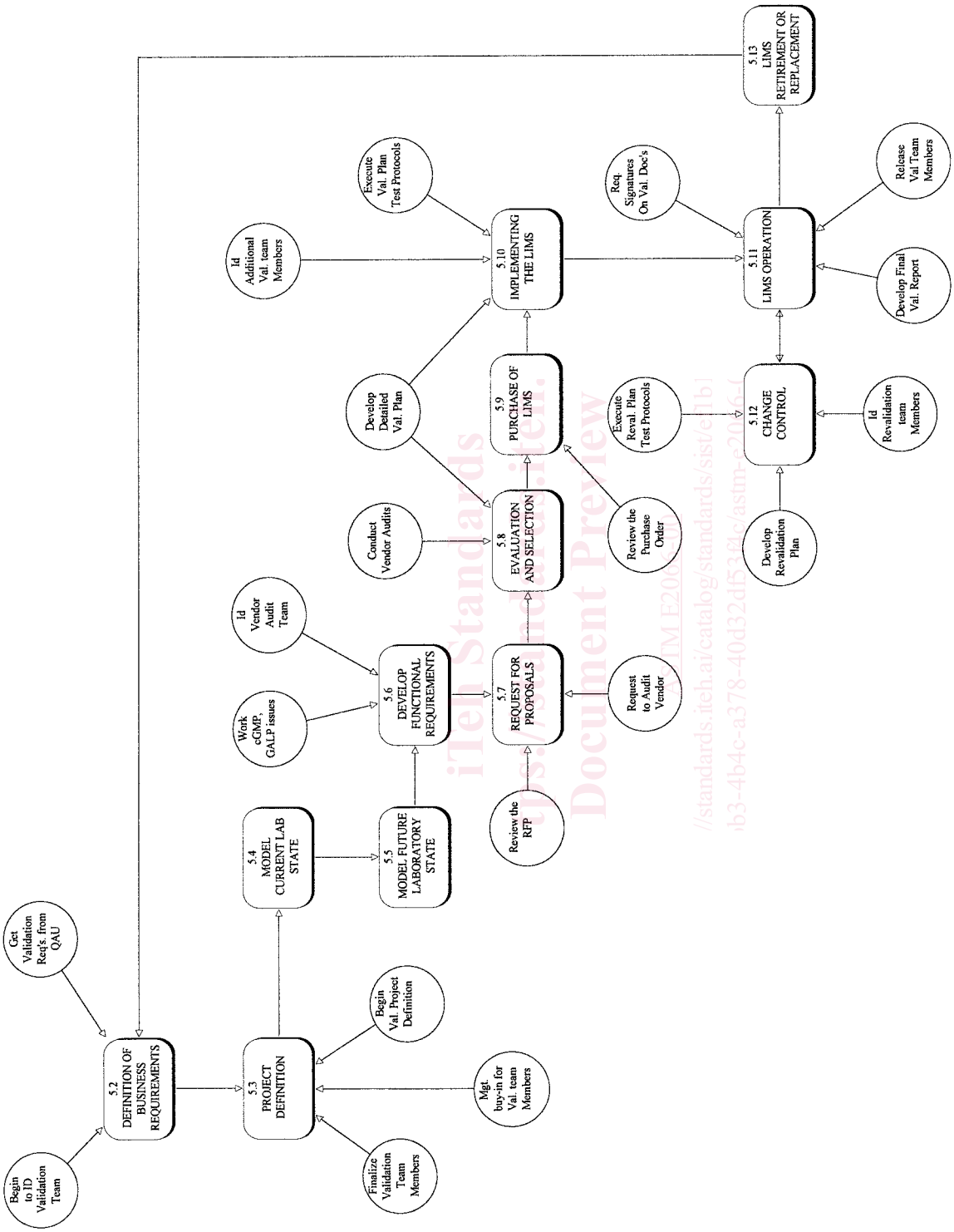

FIG. 1 LIMS Life Cycle

reviews. Since this process can take from one to several days, only those LIMS manufacturers targeted by LIMS team should be visited. The prioritized selection of LIMS shall be based upon the vendor's answers to the RFP. The RFP answers will normally emphasize the stated functional requirements. Perform a vendor audit to find the built-in quality. Continue vendor audits until an acceptable vendor for both quality and function is found. The audit results are useful in assessing the buyer's exposure to risk when system functionality is balanced against quality of system development. See Section 6 for more auditing of the LIMS vendor.

5.9 *Purchase*—Validation team members should review and be part of the purchase order approval process to ensure validation issues and criteria outlined in 5.8 are met and to begin the early stages of configuration management.

5.10 *Implementation Phase*—The validation team shall finalize the validation plan and other documentation that must be approved by the system owner and authorized by QAU before the plan is carried out. A schedule of events is developed. Testing protocols will be executed and the results documented. When all test protocols have been executed and documented, the final validation report is developed and the required signatures are obtained to approve this report. The final approval will be obtained from the system owner as authorized by QAU.

5.11 *Operational Phase*—When all validation tasks have been completed, the validation team can be disbanded. Tasks in this area include the following:

5.11.1 Ongoing training of new users.

5.11.2 Modification of SOPs to address necessary changes to the LIMS or its operational environment.

5.11.3 Review of procedures and their adherence to existing SOPs, documenting compliance with SOPs.

5.11.4 Maintenance of change control procedures for the existing system.

5.11.5 Maintenance of the system.

5.11.6 Upgrades to the LIMS hardware or software. This also includes all associated hardware or software in the LIMS operating environment, that is, the LAN, computers' operating system, etc. See the change control phase in 5.12.

5.12 *Change Control*—The LIMS Manager will face change control issues often during the normal operation of a LIMS. The LIMS Manager must understand that all minor and major changes to the system shall be subject to change control, assessment of consequences, and revalidation after the change takes place. Upgrades in software as well as changes in how the system is used may require revalidation. The change control committee may determine the system changes require revalidation. All changes shall be documented, as well as assessment of the need to validate the change and the extent of the revalidation. The level of detail for the revalidation process depends upon the type of change. A new validation team may be needed. This team may wish to include some test protocols from the original validation process. The degree of revalidation is highly dependent upon the impact of the identified change. Change requests and problems should be documented (see Appendix X6) (**3**).

5.13 *Retirement/Replacement of the LIMS*—The process starts over with the establishment of a new validation team.

6. LIMS Vendor Assessment/Audit

6.1 Industry regulators require laboratories to ensure that computer applications, such as LIMS, are validated. It is the responsibility of the laboratory owner to demonstrate that specific applications are developed, tested, operated, and maintained according to accepted quality practices.

6.2 The regulatory authorities expect that organizational personnel will follow the formal policies governing operations, as well as, comply with the proper levels of control and documentation. Further, they expect vendors to use the same level of quality control and quality practices as the customers they are supplying. It is the system owner's responsibility to investigate the vendor's operation and verify that they have accepted practices in place and that they are using them. The system owner can use the vendor audit to inspect and evaluate the vendors quality assurance programs, practices, and documentation procedures.

6.3 An organization may want to outsource vendor audits when they lack the organizational expertise, see it as a more cost effective, or they want a more objective or thorough audit. The use of audit results from a third party not associated with the user's organization, or those performed by another corporation, may not be used as a substitute for auditing the vendor. Alternatively, an audit that is jointly conducted by a consortium of corporations all looking to use a particular vendor's application has been used in the past with regulatory authority approval.

6.4 Vendor assessment should occur during the evaluation and selection phase of the LIMS life cycle and before final vendor selection. If the organization already has a vendor audit team established, this group should review their system functional requirements with the LIMS validation team. If the organization does not have such a team already established, they may want to have members of the LIMS validation team perform the vendor auditing. The audit team should be comprised of an experienced software auditor internal or external to the company and one or more individuals from the LIMS team. In general, there should be someone on the audit team responsible for the long-term relationship with the vendor. Typically, this person is the system or application owner.

6.5 The primary goal of the audit is to ensure that the vendor's software development and management procedures are consistent with the accepted practices, that is, those which are traceable back to a reference point and to which these practices adhere. This means that the audit team shall assess the vendor's quality measures, which affect the product they sell and the quality support they provide in the future. The audit team can meet this objective by gathering evidence, which demonstrates that the LIMS vendor is adhering to well-defined and documented software development and maintenance standards or practices (**4**).

6.6 In addition to these objectives, the auditing organization should evaluate the vendor's financial health and stability (**1**). It should be noted that even though a LIMS vendor organization is registered as meeting national or international requirements, for example, ISO 9001, the vendor is not exempt from

being audited by their customers. The purchasing organization is still responsible for auditing the prospective LIMS vendor. See Fig. 2 for the GAMP 96(**5**) guideline on the auditing process flowchart.

6.7 The vendor assessment should cover software development, software maintenance, quality and control issues (**4**). Key areas that should be targeted for inspection include documentation that supports system testing, preventive maintenance, operation and maintenance manuals and administrative procedures (**1**). The source code review process should be limited to a random sampling of the source code modules that the customer selects. Each item should be ranked for the

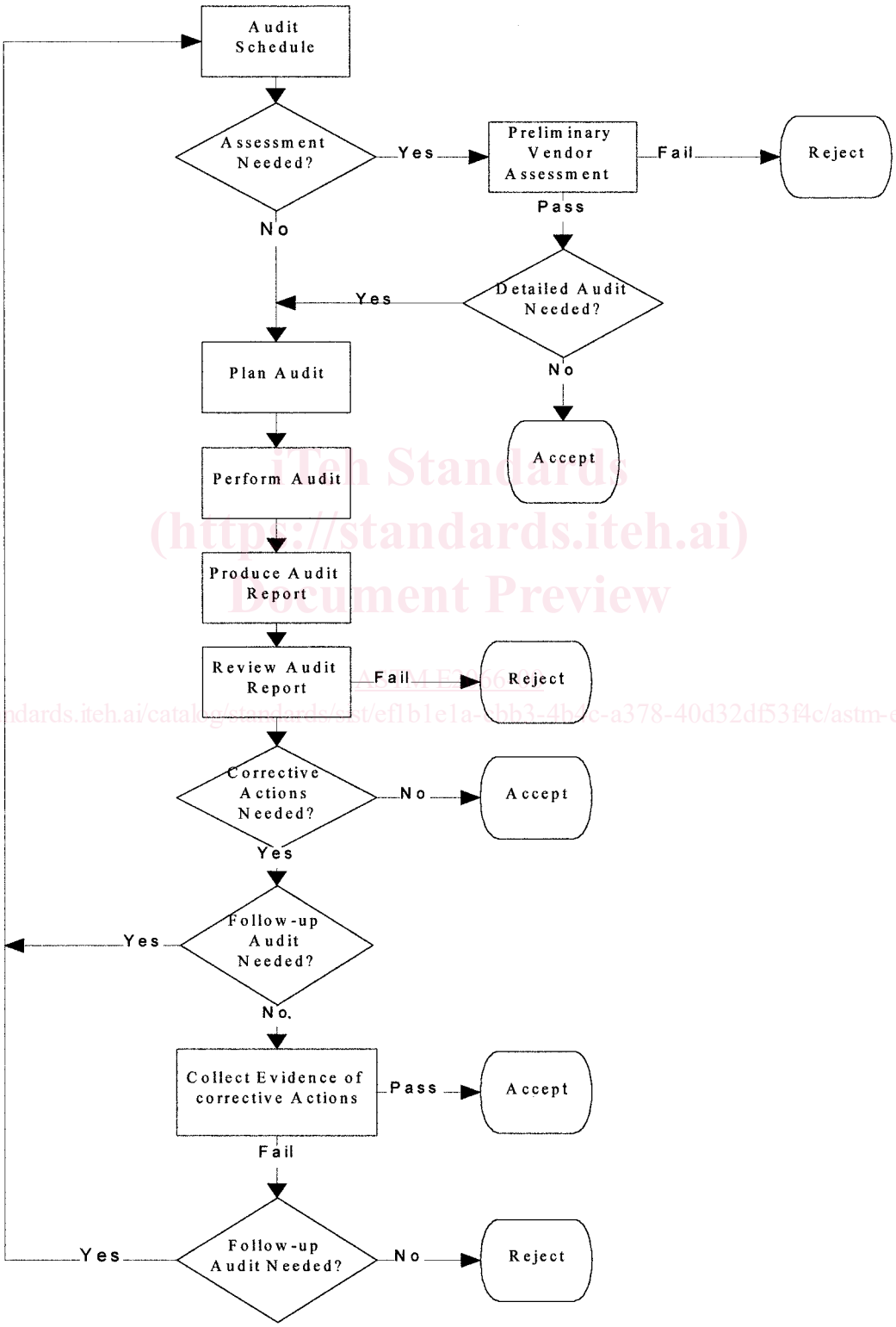

FIG. 2 Auditing Process

vendor's ability to meet that particular audit point. For example, a major discrepancy would indicate that the vendor had little or no compliance to the audit point/area. A minor discrepancy indicates that the vendor has some compliance. Both ISO 9000-3 and IEEE standards are detailed and may be used to create individualized checklists. It is important to remember that there are many different ways to accomplish compliance, and the auditor must take great care to understand how the audited company works and compare that to the standard instead of comparing it to his or her own quality system. See Appendix X1 for an overview of software items that should be investigated.

6.8 The organization should have established corporate auditing guidelines that describe in detail the procedures to which the vendor audit team shall adhere. These procedures should cover all activities from the initial vendor contact to the final meeting with the vendor. The overall auditing cycle can be divided generally into four stages: preliminary audit, detailed audit, follow-up audits, and surveillance audits (**5**). Each of these stages has its place within the overall auditing process.

6.8.1 *Preliminary Audits (Preaudit Activities)*—The goal of this stage is to gather enough documented evidence to determine if a detailed audit is required. The tool used to perform this auditing stage is typically a questionnaire. The questionnaire can be divided into the majors areas of concern, such as general corporate background information, sales information on the LIMS application (version-specific), vendor's software development life cycle (SDLC) procedures, and the product development history. Specifically, the buyer should request that the vendor supply, in advance, those standards, procedures, and plans that are associated with the LIMS application being investigated (**1**). The audit team should look for technical standards, manuals, or guides covering the following: development methodologies, software quality assurance practices, change control procedures, configuration management procedures, personnel training procedures, user support documentation, testing procedures, technical review practices, and security procedures (**1**).

6.8.2 *Detailed Audit*—When conducting these audits the organization should cover all aspects of interest relating to the application of LIMS. The validation team should plan their audit before actually performing it. The plan should establish the scope of the audit, who will be auditing, and the timing agreed to with the LIMS vendor. The audit notification should specify the purpose, timing, targeted system, scope, and the measurement criteria of the audit (**1**). The audit process itself can be divided into three major steps: the opening meeting, the review and inspection, and the closing meeting (**5**).

6.8.2.1 *Opening Meeting*—The opening meeting establishes the basic ground rules of the audit. Items to be addressed include, but are not limited to, introductions of everyone involved in this audit activity, the scope, purpose, agenda, schedule, location of the validation team meeting room, arrangements for accessing specific documents, and the signing of any confidentiality agreements by the LIMS vendor or the validation team members.

6.8.2.2 *Review and Inspection*—The audit team examines the LIMS vendor's records and their practices in accordance with these documents. The goal is to establish documented evidence that the LIMS vendor operations show adherence to their quality procedures during the LIMS development. The audit team can perform the audit using a checklist based on the scope of the audit. A successful auditor should use a "show me" approach when auditing. The required depth for coverage of each audit item will vary, but in general the audit team should identify one or two items that they will cover in great detail (**5**). The audit team may want to hold daily wrap-up sessions designed to capture that day's activities. Any observations made and their impact on quality issues should be addressed at this time. The audit team also should begin developing a list for tracking follow-up action items (**1**). This guide will aid in creation of the final audit report.

6.8.2.3 *Closing Meeting*—The lead audit team member will list all observations that the team noted during their audit. This should include positive results as well as issues of concern (**5**). The vendor's response to the observations should be included in the documentation used to develop the audit report. The audit report is important because it serves as documented evidence of the audit and its findings, as well as the basis for determining corrective actions required by the vendor. As such, the report shall present the data accurately and objectively. Because it is sensitive, the audit report should be treated as a confidential document. The audit team should close the audit with the following next steps: *(1)* the lead auditor will produce an audit report, *(2)* the audit report will be reviewed by the audit team and management, who will devise a set of corrective actions, and *(3)* the LIMS vendor should be contacted by the lead auditor and devise a plan to implement the identified corrective actions (**5**). Individuals receiving the audit report will be identified. Expected response times to address the identified weakness shall be included in the audit report (**1**).

6.8.3 *Follow-Up Audit*—Follow-up audits review the progress made by the LIMS vendor on those items identified as areas of concern on the previous audit. The organization looking to purchase the LIMS has a few options they can pursue based on the outcome of the audit report. These options include the following (**5**):

6.8.3.1 Use the LIMS supplier unconditionally.

6.8.3.2 Use the LIMS supplier for certain LIMS products only, for example, specific versions.

6.8.3.3 Use the LIMS supplier only after specific corrective actions have been carried out.

6.8.3.4 Prohibit the use of the LIMS vendor.

6.8.4 If the LIMS vendor agrees to make the necessary corrective actions outlined in the audit report, the organization purchasing the LIMS should obtain the necessary documentation from the vendor for the changes made.

6.8.5 *Surveillance Audit*—These audits focus on weaknesses found during previous audits and any new features or LIMS products, for example, a new stability study module. These audits should follow the same general guidelines adhered to by the original audit. The frequency of these audits will depend on previous audit results and criticalness of the issues that need to be addressed.

6.9 The validation team, in concert with management, should establish an action plan for those instances in which the

LIMS vendor refuses to allow an audit. The LIMS validation team must remember that it can not test quality into the system. Further, the amount of testing is proportional to the level of risk the organization will take for implementing the LIMS. Options available to the organization include the following:

6.9.1 The organization accepts the business risk and performs a much greater degree and depth of validation for the LIMS.

6.9.2 The organization rejects the LIMS vendor and moves the selection process towards alternate LIMS vendors.

7. Validation of LIMS Installed at Customer Site

7.1 The customer shall validate their use of the LIMS, independent of any vendor audit, in the operational environment in which the LIMS will be residing. The fact that a vendor's LIMS development process has been validated by the vendor or other organizations has little bearing on validating the organization's LIMS application. Further, the fact that a vendor's LIMS software has been validated by one of their other customers does not obviate the need for an organization to validate their implementation of the application.

7.2 As key functional requirements are identified and evaluated during the product evaluation phase, their results should be recorded. These results may be used in development, execution and documentation of the official LIMS test protocols. Any testing done during development of the LIMS test protocols or overall validation plan should be further refined once a specific LIMS has been selected. It should be noted that the level of testing and evaluation done during the evaluation and selection process generally will not contain enough detail to replace the test protocol used in the validation plan documentation.

7.3 The LIMS validation team may begin to identify additional resources to test the LIMS. Any new individuals selected should be familiar with the laboratory's requirements and its operation. Further, they should be knowledgeable about cGMP, GMP, GLP, GALP, or other requirements that the laboratory shall follow.

7.4 The LIMS typically is delivered as an empty database, that is, devoid of site-specific data. Configuration data and fixed laboratory information must be entered before the system can be validated. At this point, the organization starts to model their laboratory practices in LIMS. This includes test and workstation definitions and laboratory and customer personnel data. It should be noted, that during this step the laboratory may encounter additional functional requirements that were not captured initially. If the organization chooses to implement such functionality, the LIMS requirement document shall be revised to reflect these changes. Further, during this step the organization may uncover requirements that the LIMS cannot meet. The organization should document these facts and include what actions, if any, they will take to solve this problem. There are several strategies that can be used to validate a LIMS. These include, but are not limited to, the following:

7.4.1 Configure the LIMS specifically for testing with only enough configuration data to permit testing. In this case, the test system is identical to the production system, specifically, it is functioning in the same operational environment as the production system. Generally, this means that it is operating on the same computer on which the production system will reside. The configuration used in the test system shall exactly match the production system. Specifically, all reports, entry screens, queries, etc., must be identical. Furthermore, all features that are to be used in the production LIMS shall be checked for proper operation in the test system.

7.4.2 Configure the LIMS for regular operations, then isolate it from normal service while testing it. A system configured for use is called the production system. This can be accomplished by copying the production database to the test system. The LIMS program executables are the same, for example, the validation data may be part of a separate set of database tables that use the same program executables as the production LIMS, or the validation data may be part of different data group that uses the same database tables and executables as the production LIMS. The difference is in the sample data tables. If there are no problems, this approach saves time. The LIMS does not have to be configured twice, once for testing and again for production. If problems are found, partial or complete reconfiguration may be required after repairs are made. Documentation verifying that the production system is equivalent to the test system shall be provided, and the data generated during the validation process should be retained and identified as validation data.

7.4.3 A separate computer system may be used for testing.

7.4.3.1 The separate computer may be configured specifically for validation, as in 7.4.1, or it may be a copy of the production system, as in 7.4.2.

7.4.3.2 If a separate computer is used, it should have identical hardware, software, and operating system. The operating environment shall be identical to the one used for the production system. Instrument interfaces may be difficult to install on such a test system, but if they are part of the production system, they must be part of the test system as well. Ultimately, the test system could provide backup hardware for the production system.

7.4.3.3 The production and test systems may exist on the same computer, if it is sufficiently powerful, running independently. In this case, both software systems may have access to the instrument interfaces.

7.4.3.4 A subset of tests is needed when the test system is converted to the production system. These tests are used to confirm that the system still functions properly in production mode. No artificial data needs to be loaded into the active system. This subset of tests may consist of vendor-supplied diagnostic routines and little more, as long as they reliably test all parts of the proposed system. While some vendors supply these types of tools, many do not. There is no standard for their construction and execution. The use of such tools should not be the only means of testing the LIMS, but rather augment a more rigorous set of test protocols. In some cases the organization may require the tools themselves be validated prior to their use.

7.4.4 Parallel testing may be used. For a new LIMS, the manual systems can be used simultaneously with the LIMS and the results compared. If the new LIMS is a replacement, both old and new systems can be used in parallel for some period of time to compare them. The existing validated system is the

production system, while the new LIMS is the test system. Validating interfaces to instruments are an issue with a parallel testing approach, since they cannot usually be connected to both systems at the same time. In this case, the organization shall develop an approach that allows for the testing of these interfaces. The organization may want to connect these interfaces to the system undergoing validation after all other tests have been executed and just prior to the development of the final validation report. Another approach is to incorporate these interfaces as their own validation project conducted after the initial validation has been concluded.

7.5 *Response to Errors*:

7.5.1 Error handling and acceptance criteria shall be defined and described in the validation protocol and followed during the testing and reporting of results. The definition shall include criteria to be used to assess severity of errors.

7.5.2 Critical errors, such as system crashes or fatal errors, located during validation tests should be corrected or repaired immediately, before additional testing is done. Often the correction of such errors requires that most or all of the validation tests be run again. These are errors for which there is no work-around. These errors seriously threaten the integrity of the LIMS data.

7.5.3 Noncritical errors should be accumulated during the validation tests. When testing is complete, the team may decide these errors do not compromise the integrity of the information. These are errors which could result in the possibility that unacceptable result data would be accepted by the LIMS. There may be an acceptable work-around for such errors.

7.5.4 The validation team may wish to use an error grading system that helps to take action when errors are encountered. Each error would be identified by grade, and a decision would be made on what follow-up, if any, is necessary. The following are examples of grades and the errors that fall into those grades (**6**):

7.5.4.1 *Grade 0*—Typographical errors and other errors not related to the computer system.

7.5.4.2 *Grade 1*—Minor errors such as the use of upper and lower case letters used in fields not constructed for them.

7.5.4.3 *Grade 2*—Tolerable errors that must be communicated to the vendor.

7.5.4.4 *Grade 3*—Major errors that must be immediately reported to the vendor and the QA manager. All validation efforts should be suspended until QA has discussed the problem.

7.5.4.5 *Grade 4*—Disastrous errors such as relational errors in the database. These are reported the same as Grade 3 but the validation effort should be aborted. QA could still decide that the effort continue after thorough discussions.

7.6 *Standard Operating Procedures (SOPs)*:

7.6.1 SOPs are necessary for validation and ongoing operation of an organization's LIMS. These documents cover several areas, from the operation of the LIMS application through to the hardware on which the application resides. The SOPs formalize the procedures used to maintain the LIMS in a validated state by describing specific procedures to be followed. These procedure help ensure that the organization maintains a quality operation. SOPs are detailed in 11.4.

8. Validation Plan Design

8.1 The validation plan provides the overall direction of the validation process. The validation plan includes, but is not limited to, the overall objectives, a description of the system, any test boundaries or assumptions under which the validation team will be operating, the participants' responsibilities, and any general instructions for the execution of installation qualification (IQ) or operational qualification (OQ) test protocols. The validation plan needs to include a listing and description of all software and hardware components. Sometimes software modules associated with the LIMS are changed by the installation of other software. These changes could be from operating system upgrades, an upgrade to the LIMS, or other unrelated software. Further, the addition of hardware components, video cards, modems, sound cards etc., and their associated software can affect the initial LIMS validation state. The detailed listing of software and hardware components associated with the LIMS is essential as it makes up the LIMS initial configuration and describes the beginning state from which all change control is based. All test protocols for both the IQ and OQ of the associated hardware and software components are included in the validation plan. The last part of the validation plan is the signatures of the individuals responsible for ensuring that validation plan meets the organization and regulatory requirements. Typically, these signatures include the QAU validation manager, a laboratory manager, LIMS manager, and others.

8.2 IQ testing should be based on manufacturer's specifications, or recommendations, or both. Application-specific configuration will be verified as part of the IQ/OQ testing.

8.3 Vendor-supplied diagnostics can be used as part of IQ/OQ testing. IQ/OQ protocols based on vendor-supplied diagnostics shall include step-by-step verification of diagnostic procedures, recording of all results, and acceptance criteria for each result.

8.4 IQ/OQ protocol documents and test results should be produced for all hardware and software used with the LIMS, that is, operating system, database, report generators, statistical packages, network, connected instruments, computers including terminals, PCs, clients and servers, printers and plotters, bar code readers, etc. If the LIMS application is being loaded on an existing computer system, the original hardware IQ documentation may be used.

8.5 A suggested format of the IQ/OQ protocol document can be found in Appendix X2.

9. Test Protocol Design

9.1 Each organization should determine which LIMS features may attract the largest amount of attention by the auditing agencies. The organization shall determine what level of risk they are willing to accept. To validate every feature is too costly in terms of resources and time. McDowall has suggested that the organization divide the LIMS functions into one of the following three categories: must validate, should validate, and could validate (**2**).

9.1.1 The validation test protocols need to identify critical LIMS functions that will be tested. Critical LIMS functions should be based on core functions and the intended use of the

LIMS application. Rationale should be provided for not testing portions of the LIMS.

9.2 The development and execution of test protocols (TP) takes the largest amount of time in the validation effort. This fact often is overlooked when the validation project plan is developed. Many factors affect TP development and execution. First, good familiarity with the new LIMS and how it operates are essential. The less familiar the user is the longer it takes to develop detailed TPs. The validation team should build sufficient time into the project schedule for the personnel developing TPs to develop familiarity with the new system. A second factor affecting TP development is how long the TP developers have to focus upon the validation project. Not focusing enough on the TP development effort will add a significant number of additional months to the validation project. The execution of the TPs also is affected significantly by focusing the testers on the execution of the TP. A third factor affecting TP development is the number of resources available to work on the TPs. Last, the experience level of the individuals writing and executing the TPs will affect the time necessary for these activities. If possible, the organization should have at least one experienced individual working with those developing and executing the TPs.

9.3 The number of TPs necessary for validating the LIMS depends on the complexity of the LIMS and the level of detail required to adequately test the key features. TPs can be as simple as one or two lines of execution instructions or as complex as several hundred lines. The level of complexity will depend on the direction that the organization takes in the design of their TPs. Each organization should have an organizational SOP that describes how TPs are to be designed. The design can be as simple as very high level and general instructions on what testers should do and what they should expect as their acceptance criteria. TPs designed in this manner generally require the tester to write down, in detail, what they have done. At the opposite end of the spectrum are those TPs that instruct testers step by step on what to do. TPs designed in this manner typically require the testers to answer yes/no or true/false to the acceptance criteria. In either case, complex TPs can take several days to execute and document. The detail captured by testers for each TP should be sufficient enough to ensure that the LIMS function or the process being tested is under control. See Appendix X3.

9.4 In addition to execution of the TP, the validation team shall incorporate the time necessary to review TP results and to solve any identified problems. The review process can take almost as long as the execution of the TP, if the test is extremely complex. The time necessary to carry out this validation step often is underestimated. The review of each TP is necessary to ensure that the content makes sense and that it adheres to GMP documentation requirements. Specifically, all errors should have a single line drawn through them; the tester should initial, date, and give a reason why the word or group of words were crossed out. In some cases the reviewer may be responsible for deciding if the TP has met its acceptance criteria successfully, and thus, either passes or fails.

9.5 The validation team should address in the validation plan how they will handle failed TPs. This shall be addressed before the testing begins. They also should address early on how they will allow changes to the TPs after approved by the QAU. There are times when testers will need to make changes to the TP during the execution phase of a TP. Testers should be provided a way to incorporate these changes into the existing TP. The procedure shall be approved by the QAU and incorporated into the validation plan. It is essential to giver testers freedom to further design and follow additional test steps when executing the TP. This freedom allows them to explore why a particular step or set of steps did not meet its acceptance criteria. Without this freedom the entire validation project can be delayed.

9.6 All TPs shall be designed to test the given LIMS feature or function. The actual design of TPs will vary from organization to organization. The designer of the TP may wish to include any or all of the following in the design of the TP:

9.6.1 *Test Protocol Header Information*—This section contains the name of the corporation using the LIMS, the department name of the LIMS owner, date the TP was designed, statement if the TP is for IQ or OQ, TP revision number, and what system is being tested (for example, ABC LIMS Version 7.1).

9.6.2 *Test Protocol Identification Number*—Each TP should have a unique identification number. This number is only unique to the associated validation plan for the TP.

9.6.3 *Purpose*—What the TP is designed to test. For example, the purpose is to verify that new users can be added, modified, or deleted from LIMS.

9.6.4 *Requirements Under Test*—These are the functional requirements that are being tested by the TP. The TP may be designed for more than one functional requirement. Any functional requirement that was not included into the validation plan should not be included in the development of the TPs.

9.6.5 *Special Needs/Requirements*—This section lists special items that are needed to execute the TP, including specific skills the testers must have or links to other test protocols or other applications.

9.6.6 *Test Step Procedures*—Each test step should include a step number, a test procedure, and acceptance criteria for that step. Further, the test steps should be divided into and have a set of test steps for each of three categories: normal testing, stress testing, and robustness testing. Normal testing steps test the LIMS function using all common user commands. Test steps that test the function at its boundaries are stress testing. An example would be entering 20 characters into a 20 character field. Robustness testing represents testing the feature outside its boundaries. For example, a user's password may only accept character and numbers, so testers are instructed to enter special characters or punctuation characters for a newly created user's password. Testers shall identify if the test step passed of failed acceptance criteria. Typically, this is a simple yes/no statement.

9.6.7 *Comments Section*—This section is used by testers to enter their comments on any unexpected results obtained while executing the TP. Users also can capture how these unexpected results were resolved.