Database Validation, Programming and Standards

March 2009

Abstract

Success of any clinical study depends on the quality and integrity of its final database. Validation of the software system and database used for a study are crucial risk-focused quality processes for assuring and ensuring quality and integrity. This chapter discusses principles and types of validation, as well as common validation risks. Although system validation is discussed, the primary focus of the chapter is on study-specific validation, which has a greater direct impact on clinical data managers.

Introduction

The clinical data management system (CDMS) used to conduct a clinical study "…should be designed to…prevent errors in data creation, modification, maintenance, archiving, retrieval or transmission…".[1] As required by 21 CFR Part 11 and the predicate rule(s) applicable to the drug, device, or biologic in development, thorough documentation should exist at all levels of a clinical study's data collection and management. Given the multifaceted responsibilities of a CDMS, the validation process is necessary, ongoing and often complicated.

The term "validation" may refer to validation of the CDMS itself or validation of programming related to the development of a study- or protocol-specific database. Although both types of validation are crucial to the success of a study, the details of CDMS validation tend to be the responsibility of programmers or information technology (IT) personnel, although clinical data management (CDM) personnel are responsible for verifying the CDMS has been validated properly and is fit for its intended purpose.

Scope

This chapter addresses CDM validation activities that should accompany the installation of a CDMS and its patches or upgrades, as well as the testing and validation necessary when designing study-specific databases on that system. Although this chapter briefly discusses validation associated with software application development, a full description of software development validation is outside the scope of *Good Clinical Data Management Practices*. The validation measures necessary for software development and proprietary systems design are very different and more complex than the process of validating study-specific applications. This chapter also does not address validation of external data such as laboratory data. Recommendations for validation of these data are addressed in the chapter entitled "Laboratory Data Handling."

The software development life cycle (SDLC) validation approach advocated in the device and Good Laboratory Practice regulations is also appropriate for application development. The same general principles offer guidelines on the setup of individual protocols within a validated CDMS, although direct application may not always be appropriate or practical.

Although some of the specific topics addressed by this chapter may not be the direct responsibility of CDM personnel, CDM must have an ongoing awareness of the requirements and ensure these tasks have been completed in accordance with the principles and standards of their organization, regulatory bodies and good clinical practice.

Minimum Standards

- Generate a validation plan defining the testing methodology, scope, problem reporting and resolution, test data, acceptance criterion and members of the validation team.
- Ensure the CDMS meets user/functional and regulatory requirements and continues to meet these requirements through the course of its use.
- Implement the CDMS carefully, testing according to specifications, documenting all testing and issues, and ensuring objective evidence of testing is generated.

- Define processes for handling change control issues, with a clear determination of when revalidation will be required due to changes.
- Document all validation details prior to implementation in a summary document (e.g., validation report), including all applicable review and approval signatures.
- Ensure documentation remains complete and current.
- Ensure that only qualified staff develop, maintain and use the system.
- Approval of validation plan and documented results from an appropriate level of independent quality resource(s).

Best Practices

- Identify all intended user requirements of study-specific programming.
- Use organization standards, as available, to prepare study-specific programming.
- Use organization standards to document programs.
- Use code libraries wherever possible.
- Confirm that study-specific programming applications perform as intended based on the user requirements (data management plan requirements, CRF requirements, database specifications, edit check specifications, validation plan, etc.).
- Document performance during validation.
- Ensure documentation remains complete and current for live use, and is indexed for ready retrieval when it is retired or archived.
- Confirm accuracy, reliability, performance, consistency of processing and the ability to identify invalid or altered records. Confirm through testing and document.
- Ensure the system has an appropriate traceability matrix linking test cases to requirements.

- Confirm that the study-specific application has been configured properly.

Validation

"Validation" is a term applied to different processes, and is sometimes misused or used in a context that may not always be clear. Even when the term "validation" is used clearly and correctly, clear distinctions exist between validation of different systems, processes and contexts. The following descriptions distinguish between different types of validation and processes associated with validation.

- Validation versus user acceptance testing (UAT)—In *Guidance for Industry: Computerized Systems Used in Clinical Investigations*, the Food and Drug Administration (FDA) defined software validation as "Confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses and that the particular requirements implemented through the software can be consistently fulfilled."[1] UAT is one element of the examination, and documented UAT results serve as one component of "objective evidence" supporting the validation process. UAT is performed by users of the database or CDMS, and should test for both false positive and false negative results in all fields and functions. UAT does not constitute validation by itself; other elements of validation include, but are not limited to, the validation plan, requirements specifications, a traceability matrix, a UAT summary and a validation summary.

- Core CDMS validation—CDMS end users must confirm that the system has been appropriately validated prior to its release for operational use (e.g., creating individual studies). This validation is conducted using a SDLC methodology and is typically a collaboration between IT, quality assurance (QA) and end user personnel. Expected system functionality will be defined in a system requirements specification (SRS) document describing the processes followed and testing performed to ensure the product installs the way the manufacturer intended (sometimes known as installation qualification or IQ), that the system is designed according to the manufacturer's design specifications (sometimes known as operational qualification or OQ), and that the system functions according to stated requirements and the system's intended use (sometimes known as performance qualification or PQ). Primary system users should review the

results of this testing to determine if the testing has adequately demonstrated the validity of the system. Descriptions of the more prevalent types of CDMS validations are provided below:

- Commercial off-the-shelf (COTS) products—Most software developers are not directly responsible for compliance with regulatory bodies, leaving the sponsor with the ultimate responsibility for this compliance. End users should investigate and assure that the software vendor has developed and maintains the CDMS using SDLC methodology, including design level testing. This assurance can typically be provided by conducting an audit of the software vendor's development and design level validation.

- Internally developed CDMS validation—The primary distinction for an internally developed CDMS is that internal staff are responsible for developing and maintaining the CDMS. Those staff developing the CDMS should follow SDLC methodology and be held to the same standards as any vendor providing a CDMS. End users should conduct the same UAT and validation activities described in this chapter.

- Prospective CDMS validation—According to the FDA, "Prospective validation is conducted before a new product is released for distribution or, where the revisions may affect the product's characteristics, before a product made under a revised manufacturing process is released for distribution."[2] This is the type of CDMS validation most frequently performed.

- Retrospective CDMS validation—According to the FDA, "Retrospective validation is the validation of a process based on accumulated historical production, testing, control, and other information for a product already in production and distribution. This type of validation makes use of historical data and information which may be found in batch records, production log books, lot records, control charts, test and inspection results, customer complaints or lack of complaints, field failure reports, service reports, and audit reports. Historical data must contain enough information to provide an in-depth picture of how the process has been operating and whether the product has consistently met its specifications. Retrospective validation may not be feasible if all the appropriate data was not

collected, or appropriate data was not collected in a manner which allows adequate analysis."[2] Any time a CDMS must be validated while in active use, validation will be more difficult and the validation plan will be more detailed than expected for prospective validation.

- ☐ Legacy CDMS validation—Although there is no formally accepted definition for a legacy system, the term is often used to refer to a CDMS that is currently in operation but does not comply with current regulations.[3] Some may consider a legacy system one which was operational prior to the release of 21 CFR Part 11. The first step of validating a legacy system should be to perform a detailed evaluation of risks and gaps between the system and current regulations. After conducting this evaluation, CDM personnel may find the best solution is to move to a different CDMS. If the decision is made to validate the legacy system, the validation should follow the same processes and procedures as retrospective validation.

- ☐ Validation of an externally hosted CDMS—Although very similar to validation of a commercially available CDMS, validation of an externally hosted CDMS differs in that the vendor's documentation should also include information relating to infrastructure qualification, networks, servers' maintenance and logical/physical security measures.

- Study-specific validation—After a CDMS has been validated, study-specific validation must be performed to demonstrate that the requirements for the implementation of a given study using the CDMS have been successfully developed, tested and documented. The FDA states that the "Protocol should identify each step at which a computerized system will be used to create, modify, maintain, archive, retrieve, or transmit source data."[1] The processes involved with study-specific validation will be discussed in greater detail later in this chapter.

Importance of Validation to CDM

CDM plays a key role in providing high quality databases that meet both clinical and regulatory requirements. Because clinical data managed through a CDMS is the basis for the marketing approval of new drugs, devices, and biologics, it is imperative that companies seeking to market their products be

able to demonstrate the quality, reliability, reproducibility and integrity of data managed during the conduct of a clinical study. Validation provides evidence that the data management system or study-specific database meets its specifications and requirements and is therefore suitable for its intended purpose.

A clinical data manager's objective is to finish a study with a database that is accurate, secure, reliable, and ready for analysis. Any errors leading to assessment of inaccurate data may detrimentally impact the confidence of a study's results and conclusions. As stated by the FDA, "The computerized system should be designed to…prevent errors in data creation, modification, maintenance, archiving, retrieval or transmission…".[1]

SDLC and Validation

Principles of SDLC are applicable to all types of validation. One can expect that the details of each step will vary between prospective, retrospective, commercially available, internally developed, CDMS and study-specific validation, although the same general principles can be applied to each. The following are phases of SDLC and how they may apply to validation within a clinical study.

System User/Functional Requirements

Before a CDMS is designed or purchased, the requirements of the system should be clearly defined. Every organization conducting clinical research should have an SRS template listing basic IQ, OQ and PQ requirements, as well as system requirements relating to electronic records and electronic signatures as per 21 CFR Part 11.

Design and Build

For either a CDMS or study-specific database design, the process begins with a design of the program or database, which may be presented as a flowchart. Thorough documentation should be made of what the CDMS or database aims to achieve and how it will be achieved. All algorithms and programming codes should also be clearly documented.

Testing

The testing phase of the SDLC is the phase most commonly thought of as validation, although every phase is important to help ensure testing is adequate and effective. Testing should be performed at each step of development, and integrated testing should be performed to ensure all parts work together correctly once the database or CDMS is completed. A traceability matrix should be used to log which tests correspond with each SRS and to document when each test is completed.

Implementation

A database or CDMS should be put into production only after all validation activities have been completed and thoroughly documented. Once validation is completed, a final validation summary report should be produced and signed by all responsible parties.

Operation and Maintenance

Following implementation, CDM should make certain that the system continues to do what it is expected to do. This may be accomplished by maintaining thorough, appropriate documentation of records relating to training, change control/revalidations, protection of programs and data, recoverability, review of use and performance of the system, etc.

Validation Standards

Validation standards help ensure reproducibility of validation results, enhance system reliability, and ultimately increase quality. Validation standards can simplify executing the validation process by providing an assurance of the accuracy, completeness, and reliability of the CDMS or study-specific database. Validation standards ensure that each iteration of the validation process is performed consistently, thus ensuring the same level of confidence in the ongoing performance and integrity of a CDMS or study-specific database. Although standards vary between organizations, published standards from entities such as the International Organization for Standardization (ISO) and Good Automated Manufacturing Practice (GAMP) can provide a foundation for developing an organization’s validation standards.

In spite of the numerous benefits validation standards provide, they should be used in conjunction with a thorough risk assessment. Validation standards can become onerous and difficult to follow if they are inappropriately focused or scaled, but a risk assessment can help prevent CDM from doing far more work than is needed.

Validation Plan

A validation plan gives an overview of the entire validation project, describing the scope of work to be performed and processes and test procedures to be employed. The plan also describes responsibilities of different members of the validation team, which will typically consist of members of various departments, including IT, QA and CDM.

In addition to the validation plan, a validation protocol may be needed, which would be employed for software patch updates, minor software revisions or small software packages that do not warrant a full validation plan. A validation protocol will typically incorporate features of a validation plan, IQ, OQ, PQ and a traceability matrix displaying the test steps that validate each specific function.

How to Develop a Validation Plan

A validation plan clearly describes all validation activities in a manner that elucidates the plan's compliance with company, industry and regulatory standards. Some fundamental elements that should be addressed include an overview of the plan, document approval, document history, system description, roles/responsibilities, validation strategies and approaches, documentation practices, deviation/response forms, discrepancy logs and reports, a traceability matrix, a script error log, and references.

Components and Processes of a Validation Plan

A validation plan should contain the following components:

- Purpose of the validation plan
- Scope

- Project documentation development and reviews
- Schedule/timeline
- Risk analysis
- Development and test tools
- Team resources and responsibilities
- Development and test environments
- Test data sets
- Validation tasks
- Test documentation
- Test definition and execution
- Traceability matrix
- Metrics for project progress tracking
- Criteria for release into production
- Release procedures
- Required approvals
- Reporting

In addition to the components described above, the following processes should be considered:

- Validation testing
  - ☐ Test environment, test data or combination of the two
  - ☐ Manual
  - ☐ Automated
  - ☐ Metrics or quantification of validation quality criteria

- Data migration
  - ☐ Moving from one data capture system to another
  - ☐ Moving from one database to another (e.g. Access to Oracle)
  - ☐ Moving to a newer version of the same application and the appropriate revalidations that are required
- Documented processes should define when change control is appropriate
  - ☐ SOPs should say when change control is appropriate
  - ☐ SOPs should say when revalidation is appropriate
  - ☐ When and how much regression testing is appropriate?
- Validation-related risks
  - ☐ Business risk (likelihood that the system contains quality problems)
  - ☐ Audit risk (impact of negative findings from any sort of audit)

Study-Specific Validation

After a CDMS has been validated and approved for use within an organization, validation then focuses on study- or protocol-specific database design and implementation. Validation at this phase can be addressed in three major categories: database design, data entry or capture, and other study-specific programming.

Database design should be based on standard data architectures within an organization, as well as on regulatory requirements and industry standards. Utilizing standard ways of designing study databases and maintaining study data allow validation efforts and results to be easily documented, maintained, and leveraged across many projects. If data structure libraries are available, the templates should be accessible and adaptable where necessary to accommodate specific and unique project requirements. When standards are not available, efforts should be made to keep database design and data structures as consistent as possible within projects and, wherever possible, across projects. For example, data structures developed for Phase I studies

should be used throughout Phase II and III studies wherever appropriate. If use of standards is not possible, as in the case of a contract organization designing a database according to sponsor specifications, the specifications are sufficient. When designing a database according to sponsor specifications, every effort should be made to be consistent, particularly if multiple databases are designed for the same sponsor.

At a minimum, database specifications should provide the following information for each variable:

- Name and label
- Dataset label, panel, or other logical group to which the data belongs
- Type (e.g., numeric, character, integer, decimal, date)
- Length (including number of characters before and after the decimal point, where applicable)
- Definitions for all coded values included in code lists
- Algorithms for variables derived or calculated within the database or CDMS

Use of standards simplifies the specification process by providing a shorthand way of indicating standard items that are obtained from existing documentation. Some examples of standards commonly used in clinical research include those published by the Clinical Data Interchange Standards Consortium (CDISC). For more information about CDISC standards, please see http://www.cdisc.org.

When testing a study's data capture system, the most important considerations are to ensure that data entered through a data entry screen or captured via some other transfer process (e.g., electronic lab data transfers) map to the correct variables in the clinical study database and that the parameters for the variable correctly house the data provided. Useful validation measures include entering test or "dummy" data into the screens or loading test data transfer files so that output data listings and data extracted from the database can be reviewed to ensure that the variables were correctly added and saved within the database structure. Testing should be performed on all data, regardless of whether the data do or do not meet defined data structures. It is critical to

validate the data field definitions in terms of length and type. Will all study data be accepted by the database? Are variable lengths sufficient to prevent truncating or rounding? Do character and numeric formats provide the necessary output for analysis files, query management software and other modules within the sponsor's overall CDMS? If the database is programmed to flag out-of-range data, are flags appropriately triggering at data entry or import?

Database entry or capture validation testing should help identify key records management issues. For example, the database should not accept entry of duplicate records, and primary key variables should be appropriately assigned and managed by the definition of the database's structure. When discrepancies between the first and second passes of data entry are resolved for double data entry systems, validation should ensure that one record with the correct data is permanently and correctly inserted into the study database and can be extracted. Most importantly, the audit trail for the study should be validated and protected so that all manipulations of the study database or external files are recorded by date, time, and user stamps in an unalterable audit trail that can be accessed throughout the life of the data.

Other examples of study-specific programming are data loading or transfer programming (e.g., loading adverse event coding variables or loading central lab data), and programming written to validate the data (e.g., edit checks, query rules, procedures). This programming includes any code written to check the data and can occur at the time of entry or later as a batch job. This programming must be validated if action is taken regarding clinical data intended for submission as a result of the programming. Examples include programming that identifies data discrepancies such that queries are sent to clinical investigators or in-house data-editing conventions followed for items identified by the programming.

Best practices include identifying all intended uses of study-specific programming and testing each logic condition in the programming based on a validation plan. Algorithms for variable derivations occurring within the database must be validated.

Practical suggestions include utilizing organization standards to document as much of the programming specification and validation plans as possible and code libraries to reduce the amount of new code generated for a protocol. The

entire validation plan can be a standard operating procedure containing testing methodology, scope, purpose, acceptance criterion, approvals and the format for test data and problem reporting.

Validation Risks

The ultimate risk in validation is ending a study with incorrect or unreliable data, which could have a negative effect on patients' safety. There are also risks relating to relevant regulatory bodies such as the FDA. For example, regulatory bodies may not accept positive study results due to inadequate validation or validation documentation.

Validation Risks

- Scope inappropriate—Many software packages may have extraneous functionality that is not needed for the study in which the CDMS is used. Timelines and costs may dictate that only components and functions of the CDMS that will be used in the study be validated, however, any components affecting data and outcomes must be validated.

- Testing inadequate—All functional requirements must be thoroughly tested. If testing is inaccurate or incomplete, validation may not be considered successful, increasing costs and timelines by necessitating a repeat validation be performed.

- Evidence insufficient—Poor documentation is just as much a risk as inadequate testing. If auditors or inspectors are not provided with sufficient evidence to prove an adequate validation occurred, they must assume that an adequate validation did not occur. Examples of insufficient evidence include a lack of change control processes, incomplete UAT documentation or having a pass/fail checkbox without a section properly documenting the results in greater detail. In the case of validation done by a CRO for a sponsor trial, an audit should contain review of validation documentation at the CRO and a confirmation of the validation should be provided to the sponsor as part of their study documentation.

Study-Specific Validation Risks

Because study-specific programming may be perceived to have less impact than programming in a CDMS, study-specific validation may be taken for

granted by some. However, no matter how miniscule the amount of programming performed, any type of validation failure can potentially cause harm to patients' safety or the organization's bottom line. Following are some study-specific validation risks.

- User requirements not clearly defined or documented
- Incomplete testing
  - ☐ Thorough program design testing not performed prior to UAT
  - ☐ All study-specific requirements not tested
  - ☐ All edits/error messages not tested
  - ☐ All data points not tested
  - ☐ Workflows not tested
  - ☐ Challenges not robust or not performed
- Testing is inadequately documented
  - ☐ No traceability to requirements
  - ☐ Review not evident
  - ☐ Anomaly resolutions not clearly documented
  - ☐ Lack of objective evidence (e.g., screen prints) to show that the system works as intended
  - ☐ Poor organization of documentation
- Staff qualification or training not appropriate
  - ☐ Not well trained in testing protocol
  - ☐ Not familiar with business process
  - ☐ Not familiar with system

- ☐ Not familiar with applicable SOPs, testing principles, standards or conventions
- ☐ Process roles and responsibilities not well defined

- Inadequate change control processes
- Actualized risk results in financial loss (e.g., responding to inspection/audit findings, loss of clients, repeating study processes, rejected submissions)

Regulatory Impact on Validation

Those responsible for validation must be mindful of how their validation activities and documentation would be perceived in an audit or inspection by regulatory bodies. Although referring specifically to software, the following statement by the FDA could just as easily apply to study-specific validation. "Software verification and validation are difficult because a developer cannot test forever, and it is hard to know how much evidence is enough. In large measure, software validation is a matter of developing a 'level of confidence' that the device meets all requirements and user expectations for the software automated functions and features of the device. Measures such as defects found in specifications documents, estimates of defects remaining, testing coverage, and other techniques are all used to develop an acceptable level of confidence before shipping the product. The level of confidence, and therefore the level of software validation, verification, and testing effort needed, will vary depending upon the safety risk (hazard) posed by the automated functions of the device."[4]

Although the preceding quote acknowledges some of the difficulties of validation, an external audit or inspection will never say there is too much information or documentation related to system or database validation. Providing auditors or inspectors with a thorough, well-designed validation plan can help impart a comfort level that the validation has been complete and accurate.

Recommended Standard Operating Procedures

- Study-specific Database Design

- System Validation
    - ☐ UAT
    - ☐ Validation Documentation

The preceding SOPs are intended to augment the following SOPs recommended in the FDA's *Guidance for Industry: Computerized Systems Used in Clinical Investigations*, which states, "The SOPs should include, but are not limited to, the following processes.

- System setup/installation (including the description and specific use of software, hardware, and physical environment and the relationship)
- System operating manual
- Validation and functionality testing
- Data collection and handling (including data archiving, audit trails, and risk assessment)
- System maintenance (including system decommissioning)
- System security measures
- Change control
- Data backup, recovery, and contingency plans
- Alternative recording methods (in the case of system unavailability)
- Computer user training
- Roles and responsibilities of sponsors, clinical sites and other parties with respect to the use of computerized systems in the clinical trials"[1]

References

1. US Food and Drug Administration. *Guidance for Industry: Computerized Systems Used in Clinical Investigations.* Rockville, MD: US Department of Health and Human Services; 2007.

2. US Food and Drug Administration. *Guidance for Industry: Medical Device Quality Systems Manual*. Rockville, MD: US Department of Health and Human Services; 1999.

3. ISPE GAMP Forum. GAMP Good Practice Guide: The Validation of Legacy Systems. *Pharmaceutical Engineering.* November/December 2003.

4. US Food and Drug Administration. *General Principles of Software Validation; Final Guidance for Industry and FDA Staff.* Rockville, MD: US Department of Health and Human Services; 2002.

Further Reading

Association for Clinical Data Management and Statisticians in the Pharmaceutical Industry. Computer Systems Validation in Clinical Research. Macclesfield, UK: 1997.

International Conference on Harmonisation. Good Clinical Practice: Consolidated Guideline. *Federal Register*. 1997; 62(90): 2571.

Code of Federal Regulations, Title 21, Part 11, Electronic Records; Electronic Signatures. Washington, DC: US Government Printing Office; 1997.

Chapter Revision History

Publication Date	**Comments**
September 2000	Initial publication.
May 2007	Revised for style, grammar, and clarity. Substance of chapter content unchanged.
March 2009	Revised for content, style, grammar, and clarity.

This page is intentionally blank.