SoftwareCPR- Validation Training Manual
December 2008

This manual contains training references and handouts often used in our Production and Quality Systems 820.70(i) Validation and Part 11 Training.

Table of Contents

SOFTWARECPR®
CRISIS PREVENTION AND RECOVERY, LLC

www.softwarecpr.com
20 Berkshire Drive
Winchester, MA 01890
781-721-2921

Copyright

Legal Disclaimer

The training document that follows should only be applied in the appropriate context with oversight by regulatory and software professionals with direct knowledge and experience with the topics presented. The document should not be used as a cookbook since it is not adequate for some situations and may be excessive for others.

While SoftwareCPR® attempts to ensure the accuracy of information presented, no guarantees are made since regulatory interpretations and enforcement practices are constantly changing, and are not entirely uniform in their application.

Disclaimer of Warranties: The information is provided AS IS, without warranties of any kind. CPRLLC does not represent or warrant that any information or data provided herein is suitable for a particular purpose. CPRLLC hereby disclaims and negates any and all warranties, whether express or implied, relating to such information and data, including the warranties of merchantability and fitness for a particular purpose.

General Principles of Software Validation; Final Guidance for Industry and FDA Staff

Document issued on: January 11, 2002

This document supersedes the draft document, "General Principles of Software Validation, Version 1.1, dated June 9, 1997.

U.S. Department Of Health and Human Services
Food and Drug Administration
Center for Devices and Radiological Health
Center for Biologics Evaluation and Research

Preface

Public Comment

Comments and suggestions may be submitted at any time for Agency consideration to Dockets Management Branch, Division of Management Systems and Policy, Office of Human Resources and Management Services, Food and Drug Administration, 5630 Fishers Lane, Room 1061, (HFA-305), Rockville, MD, 20852. When submitting comments, please refer to the exact title of this guidance document. Comments may not be acted upon by the Agency until the document is next revised or updated.

For questions regarding the use or interpretation of this guidance which involve the Center for Devices and Radiological Health (CDRH), contact John F. Murray at (301) 594-4659 or email jfm@cdrh.fda.gov

For questions regarding the use or interpretation of this guidance which involve the Center for Biologics Evaluation and Research (CBER) contact Jerome Davis at (301) 827-6220 or email davis@cber.fda.gov.

Additional Copies

CDRH
Additional copies are available from the Internet at: http://www.fda.gov/cdrh/comp/guidance/938.pdf or via CDRH Facts-On-Demand. In order to receive this document via your fax machine, call the CDRH Facts-On-Demand system at 800-899-0381 or 301-827-0111 from a touch-tone telephone. Press 1 to enter the system. At the second voice prompt, press 1 to order a document. Enter the document number 938 followed by the pound sign (#). Follow the remaining voice prompts to complete your request.

CBER
Additional copies are available from the Internet at: http://www.fda.gov/cber/guidelines.htm, by writing to CBER, Office of Communication, Training, and Manufacturers' Assistance (HFM-40), 1401 Rockville Pike, Rockville, Maryland 20852-1448, or by telephone request at 1-800-835-5709 or 301-827-1800.

Table of Contents

General Principles of Software Validation

This document is intended to provide guidance. It represents the Agency's current thinking on this topic. It does not create or confer any rights for or on any person and does not operate to bind Food and Drug Administration (FDA) or the public. An alternative approach may be used if such approach satisfies the requirements of the applicable statutes and regulations.

SECTION 1. PURPOSE

This guidance outlines general validation principles that the Food and Drug Administration (FDA) considers to be applicable to the validation of medical device software or the validation of software used to design, develop, or manufacture medical devices. This final guidance document, Version 2.0, supersedes the draft document, *General Principles of Software Validation, Version 1.1*, dated June 9, 1997.

SECTION 2. SCOPE

This guidance describes how certain provisions of the medical device Quality System regulation apply to software and the agency's current approach to evaluating a software validation system. For example, this document lists elements that are acceptable to the FDA for the validation of software; however, it does not list all of the activities and tasks that must, in all instances, be used to comply with the law.

The scope of this guidance is somewhat broader than the scope of validation in the strictest definition of that term. Planning, verification, testing, traceability, configuration management, and many other aspects of good software engineering discussed in this guidance are important activities that together help to support a final conclusion that software is validated.

This guidance recommends an integration of software life cycle management and risk management activities. Based on the intended use and the safety risk associated with the software to be developed, the software developer should determine the specific approach, the combination of techniques to be used, and the level of effort to be applied. While this guidance does not recommend any specific life cycle model or any specific technique or method, it does recommend that software validation and verification activities be conducted throughout the entire software life cycle.

Where the software is developed by someone other than the device manufacturer (e.g., off-the-shelf software) the software developer may not be directly responsible for compliance with FDA regulations.

In that case, the party with regulatory responsibility (i.e., the device manufacturer) needs to assess the adequacy of the off-the-shelf software developer's activities and determine what additional efforts are needed to establish that the software is validated for the device manufacturer's intended use.

2.1. APPLICABILITY

This guidance applies to:

- Software used as a component, part, or accessory of a medical device;
- Software that is itself a medical device (e.g., blood establishment software);
- Software used in the production of a device (e.g., programmable logic controllers in manufacturing equipment); and
- Software used in implementation of the device manufacturer's quality system (e.g., software that records and maintains the device history record).

This document is based on generally recognized software validation principles and, therefore, can be applied to any software. For FDA purposes, this guidance applies to any software related to a regulated medical device, as defined by Section 201(h) of the Federal Food, Drug, and Cosmetic Act (the Act) and by current FDA software and regulatory policy. This document does not specifically identify which software is or is not regulated.

2.2. AUDIENCE

This guidance provides useful information and recommendations to the following individuals:

- Persons subject to the medical device Quality System regulation
- Persons responsible for the design, development, or production of medical device software
- Persons responsible for the design, development, production, or procurement of automated tools used for the design, development, or manufacture of medical devices or software tools used to implement the quality system itself
- FDA Investigators
- FDA Compliance Officers
- FDA Scientific Reviewers

2.3. THE LEAST BURDENSOME APPROACH

We believe we should consider the least burdensome approach in all areas of medical device regulation. This guidance reflects our careful review of the relevant scientific and legal requirements and what we believe is the least burdensome way for you to comply with those requirements. However, if you believe that an alternative approach would be less burdensome, please contact us so we can consider your point of view. You may send your written comments to the contact person listed in the preface to this guidance or to the CDRH Ombudsman. Comprehensive information on CDRH's Ombudsman, including ways to contact him, can be found on the Internet at:

http://www.fda.gov/cdrh/resolvingdisputes/ombudsman.html.

2.4. REGULATORY REQUIREMENTS FOR SOFTWARE VALIDATION

The FDA's analysis of 3140 medical device recalls conducted between 1992 and 1998 reveals that 242 of them (7.7%) are attributable to software failures. Of those software related recalls, 192 (or 79%) were caused by software defects that were introduced when changes were made to the software after its initial production and distribution. Software validation and other related good software engineering practices discussed in this guidance are a principal means of avoiding such defects and resultant recalls.

Software validation is a requirement of the Quality System regulation, which was published in the Federal Register on October 7, 1996 and took effect on June 1, 1997. (See Title 21 Code of Federal Regulations (CFR) Part 820, and 61 Federal Register (FR) 52602, respectively.) Validation requirements apply to software used as components in medical devices, to software that is itself a medical device, and to software used in production of the device or in implementation of the device manufacturer's quality system.

Unless specifically exempted in a classification regulation, any medical device software product developed after June 1, 1997, regardless of its device class, is subject to applicable design control provisions. (See of 21 CFR §820.30.) This requirement includes the completion of current development projects, all new development projects, and all changes made to existing medical device software. Specific requirements for validation of device software are found in 21 CFR §820.30(g). Other design controls, such as planning, input, verification, and reviews, are required for medical device software. (See 21 CFR §820.30.) The corresponding documented results from these activities can provide additional support for a conclusion that medical device software is validated.

Any software used to automate any part of the device production process or any part of the quality system must be validated for its intended use, as required by 21 CFR §820.70(i). This requirement applies to any software used to automate device design, testing, component acceptance, manufacturing, labeling, packaging, distribution, complaint handling, or to automate any other aspect of the quality system.

In addition, computer systems used to create, modify, and maintain electronic records and to manage electronic signatures are also subject to the validation requirements. (See 21 CFR §11.10(a).) Such computer systems must be validated to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records.

Software for the above applications may be developed in-house or under contract. However, software is frequently purchased off-the-shelf for a particular intended use. All production and/or quality system software, even if purchased off-the-shelf, should have documented requirements that fully define its intended use, and information against which testing results and other evidence can be compared, to show that the software is validated for its intended use.

The use of off-the-shelf software in automated medical devices and in automated manufacturing and quality system operations is increasing. Off-the-shelf software may have many capabilities, only a few of which are needed by the device manufacturer. Device manufacturers are responsible for the adequacy of the software used in their devices, and used to produce devices. When device manufacturers purchase "off-the-shelf" software, they must ensure that it will perform as intended in their chosen application. For off-the-shelf software used in manufacturing or in the quality system, additional guidance is included in Section 6.3 of this document. For device software, additional useful information may be found in FDA's *Guidance for Industry, FDA Reviewers, and Compliance on Off-The-Shelf Software Use in Medical Devices*.

2.4. QUALITY SYSTEM REGULATION VS PRE-MARKET SUBMISSIONS

This document addresses Quality System regulation issues that involve the implementation of software validation. It provides guidance for the management and control of the software validation process. The management and control of the software validation process should not be confused with any other validation requirements, such as process validation for an automated manufacturing process.

Device manufacturers may use the same procedures and records for compliance with quality system and design control requirements, as well as for pre-market submissions to FDA. This document does not cover any specific safety or efficacy issues related to software validation. Design issues and documentation requirements for pre-market submissions of regulated software are not addressed by this document. Specific issues related to safety and efficacy, and the documentation required in pre-market submissions, should be addressed to the Office of Device Evaluation (ODE), Center for Devices and Radiological Health (CDRH) or to the Office of Blood Research and Review, Center for Biologics Evaluation and Research (CBER). See the references in Appendix A for applicable FDA guidance documents for pre-market submissions.

SECTION 3. CONTEXT FOR SOFTWARE VALIDATION

Many people have asked for specific guidance on what FDA expects them to do to ensure compliance with the Quality System regulation with regard to software validation. Information on software validation presented in this document is not new. Validation of software, using the principles and tasks listed in Sections 4 and 5, has been conducted in many segments of the software industry for well over 20 years.

Due to the great variety of medical devices, processes, and manufacturing facilities, it is not possible to state in one document all of the specific validation elements that are applicable. However, a general application of several broad concepts can be used successfully as guidance for software validation. These broad concepts provide an acceptable framework for building a comprehensive approach to software validation. Additional specific information is available from many of the references listed in Appendix A.

3.1. DEFINITIONS AND TERMINOLOGY

Unless defined in the Quality System regulation, or otherwise specified below, all other terms used in this guidance are as defined in the current edition of the FDA *Glossary of Computerized System and Software Development Terminology*.

The medical device Quality System regulation (21 CFR 820.3(k)) defines "**establish**" to mean "define, document, and implement." Where it appears in this guidance, the words "establish" and “established” should be interpreted to have this same meaning.

Some definitions found in the medical device Quality System regulation can be confusing when compared to commonly used terminology in the software industry. Examples are requirements, specification, verification, and validation.

3.1.1 Requirements and Specifications

While the Quality System regulation states that design input requirements must be documented, and that specified requirements must be verified, the regulation does not further clarify the distinction between the terms “requirement” and “specification.” A **requirement** can be any need or expectation for a system or for its software. Requirements reflect the stated or implied needs of the customer, and may be market-based, contractual, or statutory, as well as an organization's internal requirements. There can be many different kinds of requirements (e.g., design, functional, implementation, interface, performance, or physical requirements). Software requirements are typically derived from the system requirements for those aspects of system functionality that have been allocated to software. Software requirements are typically stated in functional terms and are defined, refined, and updated as a development project progresses. Success in accurately and completely documenting software requirements is a crucial factor in successful validation of the resulting software.

A **specification** is defined as “a document that states requirements.” (See 21 CFR §820.3(y).) It may refer to or include drawings, patterns, or other relevant documents and usually indicates the means and the criteria whereby conformity with the requirement can be checked. There are many different kinds of written specifications, e.g., system requirements specification, software requirements specification, software design specification, software test specification, software integration specification, etc. All of these documents establish “specified requirements” and are design outputs for which various forms of verification are necessary.

3.1.2 Verification and Validation

The Quality System regulation is harmonized with *ISO 8402*:1994, which treats “verification” and “validation” as separate and distinct terms. On the other hand, many software engineering journal articles and textbooks use the terms "verification" and "validation" interchangeably, or in some cases refer to software "verification, validation, and testing (VV&T)" as if it is a single concept, with no distinction among the three terms.

Software verification provides objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase. Software verification looks for consistency, completeness, and correctness of the software and its supporting documentation, as it is being developed, and provides support for a subsequent conclusion that software is validated. Software testing is one of many verification activities intended to confirm that software development output meets its input requirements. Other verification activities include various static and dynamic analyses, code and document inspections, walkthroughs, and other techniques.

Software validation is a part of the design validation for a finished device, but is not separately defined in the Quality System regulation. For purposes of this guidance, FDA considers software validation to be “**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 software can be consistently fulfilled.**” In practice, software validation activities may occur both during, as well as at the end of the software development life cycle to ensure that all requirements have been fulfilled. Since software is usually part of a larger hardware system, the validation of software typically includes evidence that all software requirements have been implemented correctly and completely and are traceable to system requirements. A conclusion that software is validated is highly dependent upon comprehensive software testing, inspections, analyses, and other verification tasks performed at each stage of the software development life cycle. Testing of device software functionality in a simulated use environment, and user site testing are typically included as components of an overall design validation program for a software automated device.

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. Additional guidance regarding safety risk management for software may be found in Section 4 of FDA's *Guidance for the Content of Pre-market Submissions for Software Contained in Medical Devices*, and in the international standards *ISO/IEC 14971-1* and *IEC 60601-1-4* referenced in Appendix A.

3.1.3 IQ/OQ/PQ

For many years, both FDA and regulated industry have attempted to understand and define software validation within the context of process validation terminology. For example, industry documents and other FDA validation guidance sometimes describe user site software validation in terms of installation qualification (IQ), operational qualification (OQ) and performance qualification (PQ). Definitions of these terms and additional information regarding IQ/OQ/PQ may be found in FDA's *Guideline on General Principles of Process Validation*, dated May 11, 1987, and in FDA's *Glossary of Computerized System and Software Development Terminology*, dated August 1995.

While IQ/OQ/PQ terminology has served its purpose well and is one of many legitimate ways to organize software validation tasks at the user site, this terminology may not be well understood among many software professionals, and it is not used elsewhere in this document. However, both FDA personnel and device manufacturers need to be aware of these differences in terminology as they ask for and provide information regarding software validation.

3.2. SOFTWARE DEVELOPMENT AS PART OF SYSTEM DESIGN

The decision to implement system functionality using software is one that is typically made during system design. Software requirements are typically derived from the overall system requirements and design for those aspects in the system that are to be implemented using software. There are user needs and intended uses for a finished device, but users typically do not specify whether those requirements are to be met by hardware, software, or some combination of both. Therefore, software validation must be considered within the context of the overall design validation for the system.

A documented requirements specification represents the user's needs and intended uses from which the product is developed. A primary goal of software validation is to then demonstrate that all completed software products comply with all documented software and system requirements. The correctness and completeness of both the system requirements and the software requirements should be addressed as part of the design validation process for the device. Software validation includes confirmation of conformance to all software specifications and confirmation that all software requirements are traceable to the system specifications. Confirmation is an important part of the overall design validation to ensure that all aspects of the medical device conform to user needs and intended uses.

3.3. SOFTWARE IS DIFFERENT FROM HARDWARE

While software shares many of the same engineering tasks as hardware, it has some very important differences. For example:

- The vast majority of software problems are traceable to errors made during the design and development process. While the quality of a hardware product is highly dependent on design, development and manufacture, the quality of a software product is dependent primarily on design and development with a minimum concern for software manufacture. Software manufacturing consists of reproduction that can be easily verified. It is not difficult to manufacture thousands of program copies that function exactly the same as the original; the difficulty comes in getting the original program to meet all specifications.

- One of the most significant features of software is branching, i.e., the ability to execute alternative series of commands, based on differing inputs. This feature is a major contributing factor for another characteristic of software – its complexity. Even short programs can be very complex and difficult to fully understand.

- Typically, testing alone cannot fully verify that software is complete and correct. In addition to testing, other verification techniques and a structured and documented development process should be combined to ensure a comprehensive validation approach.

- Unlike hardware, software is not a physical entity and does not wear out. In fact, software may improve with age, as latent defects are discovered and removed. However, as software is constantly updated and changed, such improvements are sometimes countered by new defects introduced into the software during the change.

- Unlike some hardware failures, software failures occur without advanced warning. The software's branching that allows it to follow differing paths during execution, may hide some latent defects until long after a software product has been introduced into the marketplace.

- Another related characteristic of software is the speed and ease with which it can be changed. This factor can cause both software and non-software professionals to believe that software problems can be corrected easily. Combined with a lack of understanding of software, it can lead managers to believe that tightly controlled engineering is not needed as much for software as it is for hardware. In fact, the opposite is true. **Because of its complexity, the development process for software should be even more tightly controlled than for hardware, in order to prevent problems that cannot be easily detected later in the development process**.

- Seemingly insignificant changes in software code can create unexpected and very significant problems elsewhere in the software program. The software development process should be sufficiently well planned, controlled, and documented to detect and correct unexpected results from software changes.

- Given the high demand for software professionals and the highly mobile workforce, the software personnel who make maintenance changes to software may not have been involved in the original software development. Therefore, accurate and thorough documentation is essential.

- Historically, software components have not been as frequently standardized and interchangeable as hardware components. However, medical device software developers are beginning to use component-based development tools and techniques. Object-oriented methodologies and the use of off-the-shelf software components hold promise for faster and less expensive software development. However, component-based approaches require very careful attention during integration. Prior to integration, time is needed to fully define and develop reusable software code and to fully understand the behavior of off-the-shelf components.

For these and other reasons, software engineering needs an even greater level of managerial scrutiny and control than does hardware engineering.

3.4. BENEFITS OF SOFTWARE VALIDATION

Software validation is a critical tool used to assure the quality of device software and software automated operations. Software validation can increase the usability and reliability of the device, resulting in decreased failure rates, fewer recalls and corrective actions, less risk to patients and users, and reduced liability to device manufacturers. Software validation can also reduce long term costs by making it easier and less costly to reliably modify software and revalidate software changes. Software maintenance can represent a very large percentage of the total cost of software over its entire life cycle. An established comprehensive software validation process helps to reduce the long-term cost of software by reducing the cost of validation for each subsequent release of the software.

3.5 DESIGN REVIEW

Design reviews are documented, comprehensive, and systematic examinations of a design to evaluate the adequacy of the design requirements, to evaluate the capability of the design to meet these requirements, and to identify problems. While there may be many informal technical reviews that occur within the development team during a software project, a formal design review is more structured and includes participation from others outside the development team. Formal design reviews may reference or include results from other formal and informal reviews. Design reviews may be conducted separately for the software, after the software is integrated with the hardware into the system, or both. Design reviews should include examination of development plans, requirements specifications, design specifications, testing plans and procedures, all other documents and activities associated with the project, verification results from each stage of the defined life cycle, and validation results for the overall device.

Design review is a primary tool for managing and evaluating development projects. For example, formal design reviews allow management to confirm that all goals defined in the software validation plan have been achieved. The Quality System regulation requires that at least one formal design review be conducted during the device design process. However, it is recommended that multiple design reviews be conducted (e.g., at the end of each software life cycle activity, in preparation for proceeding to the next activity). Formal design review is especially important at or near the end of the requirements activity, before major resources have been committed to specific design solutions. Problems found at this point can be resolved more easily, save time and money, and reduce the likelihood of missing a critical issue.

Answers to some key questions should be documented during formal design reviews. These include:

- Have the appropriate tasks and expected results, outputs, or products been established for each software life cycle activity?

- Do the tasks and expected results, outputs, or products of each software life cycle activity:

  ✓ Comply with the requirements of other software life cycle activities in terms of correctness, completeness, consistency, and accuracy?

  ✓ Satisfy the standards, practices, and conventions of that activity?

  ✓ Establish a proper basis for initiating tasks for the next software life cycle activity?

SECTION 4. PRINCIPLES OF SOFTWARE VALIDATION

This section lists the general principles that should be considered for the validation of software.

4.1. REQUIREMENTS

A documented software requirements specification provides a baseline for both validation and verification. The software validation process cannot be completed without an established software requirements specification (Ref: 21 CFR 820.3(z) and (aa) and 820.30(f) and (g)).

4.2. DEFECT PREVENTION

Software quality assurance needs to focus on preventing the introduction of defects into the software development process and not on trying to "test quality into" the software code after it is written. Software testing is very limited in its ability to surface all latent defects in software code. For example, the complexity of most software prevents it from being exhaustively tested. **Software testing is a necessary activity. However, in most cases software testing by itself is not sufficient to establish confidence that the software is fit for its intended use.** In order to establish that confidence, software developers should use a mixture of methods and techniques to prevent software errors and to detect software errors that do occur. The "best mix" of methods depends on many factors including the development environment, application, size of project, language, and risk.

4.3. TIME AND EFFORT

To build a case that the software is validated requires time and effort. Preparation for software validation should begin early, i.e., during design and development planning and design input. The final conclusion that the software is validated should be based on evidence collected from planned efforts conducted throughout the software lifecycle.

4.4. SOFTWARE LIFE CYCLE

Software validation takes place within the environment of an established software life cycle. The software life cycle contains software engineering tasks and documentation necessary to support the software validation effort. In addition, the software life cycle contains specific verification and validation tasks that are appropriate for the intended use of the software. This guidance does not recommend any particular life cycle models – only that they should be selected and used for a software development project.

4.5. PLANS

The software validation process is defined and controlled through the use of a plan. The software validation plan defines "what" is to be accomplished through the software validation effort. Software validation plans are a significant quality system tool. Software validation plans specify areas such as scope, approach, resources, schedules and the types and extent of activities, tasks, and work items.

4.6. PROCEDURES

The software validation process is executed through the use of procedures. These procedures establish "how" to conduct the software validation effort. The procedures should identify the specific actions or sequence of actions that must be taken to complete individual validation activities, tasks, and work items.

4.7. SOFTWARE VALIDATION AFTER A CHANGE

Due to the complexity of software, a seemingly small local change may have a significant global system impact. When any change (even a small change) is made to the software, the validation status of the software needs to be re-established. **Whenever software is changed, a validation analysis should be conducted not just for validation of the individual change, but also to determine the extent and impact of that change on the entire software system.** Based on this analysis, the software developer should then conduct an appropriate level of software regression testing to show that unchanged but vulnerable portions of the system have not been adversely affected. Design controls and appropriate regression testing provide the confidence that the software is validated after a software change.

4.8. VALIDATION COVERAGE

Validation coverage should be based on the software's complexity and safety risk – not on firm size or resource constraints. The selection of validation activities, tasks, and work items should be commensurate with the complexity of the software design and the risk associated with the use of the software for the specified intended use. For lower risk devices, only baseline validation activities may be conducted. As the risk increases additional validation activities should be added to cover the additional risk. Validation documentation should be sufficient to demonstrate that all software validation plans and procedures have been completed successfully.

4.9. INDEPENDENCE OF REVIEW

Validation activities should be conducted using the basic quality assurance precept of "independence of review." Self-validation is extremely difficult. When possible, an independent evaluation is always better, especially for higher risk applications. Some firms contract out for a third-party independent verification and validation, but this solution may not always be feasible. Another approach is to assign internal staff members that are not involved in a particular design or its implementation, but who have sufficient knowledge to evaluate the project and conduct the verification and validation activities. Smaller firms may need to be creative in how tasks are organized and assigned in order to maintain internal independence of review.

4.10. FLEXIBILITY AND RESPONSIBILITY

Specific implementation of these software validation principles may be quite different from one application to another. The device manufacturer has flexibility in choosing how to apply these validation principles, but retains ultimate responsibility for demonstrating that the software has been validated.

Software is designed, developed, validated, and regulated in a wide spectrum of environments, and for a wide variety of devices with varying levels of risk. FDA regulated medical device applications include software that:

- Is a component, part, or accessory of a medical device;
- Is itself a medical device; or
- Is used in manufacturing, design and development, or other parts of the quality system.

In each environment, software components from many sources may be used to create the application (e.g., in-house developed software, off-the-shelf software, contract software, shareware). In addition, software components come in many different forms (e.g., application software, operating systems, compilers, debuggers, configuration management tools, and many more). The validation of software in these environments can be a complex undertaking; therefore, it is appropriate that all of these software validation principles be considered when designing the software validation process. The resultant software validation process should be commensurate with the safety risk associated with the system, device, or process.

Software validation activities and tasks may be dispersed, occurring at different locations and being conducted by different organizations. However, regardless of the distribution of tasks, contractual relations, source of components, or the development environment, the device manufacturer or specification developer retains ultimate responsibility for ensuring that the software is validated.

SECTION 5. ACTIVITIES AND TASKS

Software validation is accomplished through a series of activities and tasks that are planned and executed at various stages of the software development life cycle. These tasks may be one time occurrences or may be iterated many times, depending on the life cycle model used and the scope of changes made as the software project progresses.

5.1. SOFTWARE LIFE CYCLE ACTIVITIES

This guidance does not recommend the use of any specific software life cycle model. Software developers should establish a software life cycle model that is appropriate for their product and organization. The software life cycle model that is selected should cover the software from its birth to its retirement. Activities in a typical software life cycle model include the following:

- Quality Planning
- System Requirements Definition
- Detailed Software Requirements Specification
- Software Design Specification
- Construction or Coding
- Testing
- Installation
- Operation and Support
- Maintenance
- Retirement

Verification, testing, and other tasks that support software validation occur during each of these activities. A life cycle model organizes these software development activities in various ways and provides a framework for monitoring and controlling the software development project. Several software life cycle models (e.g., waterfall, spiral, rapid prototyping, incremental development, etc.) are defined in FDA's *Glossary of Computerized System and Software Development Terminology*, dated August 1995. These and many other life cycle models are described in various references listed in Appendix A.

5.2. TYPICAL TASKS SUPPORTING VALIDATION

For each of the software life cycle activities, there are certain "typical" tasks that support a conclusion that the software is validated. However, the specific tasks to be performed, their order of performance, and the iteration and timing of their performance will be dictated by the specific software life cycle model that is selected and the safety risk associated with the software application. For very low risk applications, certain tasks may not be needed at all. However, the software developer should at least consider each of these tasks and should define and document which tasks are or are not appropriate for their specific application. The following discussion is generic and is not intended to prescribe any particular software life cycle model or any particular order in which tasks are to be performed.

5.2.1. Quality Planning

Design and development planning should culminate in a plan that identifies necessary tasks, procedures for anomaly reporting and resolution, necessary resources, and management review requirements, including formal design reviews. A software life cycle model and associated activities should be identified, as well as those tasks necessary for each software life cycle activity. The plan should include:

- The specific tasks for each life cycle activity;
- Enumeration of important quality factors (e.g., reliability, maintainability, and usability);
- Methods and procedures for each task;
- Task acceptance criteria;
- Criteria for defining and documenting outputs in terms that will allow evaluation of their conformance to input requirements;
- Inputs for each task;
- Outputs from each task;
- Roles, resources, and responsibilities for each task;
- Risks and assumptions; and
- Documentation of user needs.

Management must identify and provide the appropriate software development environment and resources. (See 21 CFR §820.20(b)(1) and (2).) Typically, each task requires personnel as well as physical resources. The plan should identify the personnel, the facility and equipment resources for each task, and the role that risk (hazard) management will play. A configuration management plan should be developed that will guide and control multiple parallel development activities and ensure proper communications and documentation. Controls are necessary to ensure positive and correct correspondence among all approved versions of the specifications documents, source code, object code, and test suites that comprise a software system. The controls also should ensure accurate identification of, and access to, the currently approved versions.

Procedures should be created for reporting and resolving software anomalies found through validation or other activities. Management should identify the reports and specify the contents, format, and responsible organizational elements for each report. Procedures also are necessary for the review and approval of software development results, including the responsible organizational elements for such reviews and approvals.

<u>Typical Tasks – Quality Planning</u>

- Risk (Hazard) Management Plan
- Configuration Management Plan

- Software Quality Assurance Plan
  - Software Verification and Validation Plan
    - ❑ Verification and Validation Tasks, and Acceptance Criteria
    - ❑ Schedule and Resource Allocation (for software verification and validation activities)
    - ❑ Reporting Requirements
  - Formal Design Review Requirements
  - Other Technical Review Requirements
- Problem Reporting and Resolution Procedures
- Other Support Activities

5.2.2. Requirements

Requirements development includes the identification, analysis, and documentation of information about the device and its intended use. Areas of special importance include allocation of system functions to hardware/software, operating conditions, user characteristics, potential hazards, and anticipated tasks. In addition, the requirements should state clearly the intended use of the software.

The software requirements specification document should contain a written definition of the software functions. It is not possible to validate software without predetermined and documented software requirements. Typical software requirements specify the following:

- All software system inputs;
- All software system outputs;
- All functions that the software system will perform;
- All performance requirements that the software will meet, (e.g., data throughput, reliability, and timing);
- The definition of all external and user interfaces, as well as any internal software-to-system interfaces;
- How users will interact with the system;
- What constitutes an error and how errors should be handled;
- Required response times;
- The intended operating environment for the software, if this is a design constraint (e.g., hardware platform, operating system);
- All ranges, limits, defaults, and specific values that the software will accept; and
- All safety related requirements, specifications, features, or functions that will be implemented in software.

Software safety requirements are derived from a technical risk management process that is closely integrated with the system requirements development process. Software requirement specifications should identify clearly the potential hazards that can result from a software failure in the system as well as any safety requirements to be implemented in software. The consequences of software failure should be evaluated, along with means of mitigating such failures (e.g., hardware mitigation, defensive programming, etc.). From this analysis, it should be possible to identify the most appropriate measures necessary to prevent harm.

The Quality System regulation requires a mechanism for addressing incomplete, ambiguous, or conflicting requirements. (See 21 CFR 820.30(c).) Each requirement (e.g., hardware, software, user, operator interface, and safety) identified in the software requirements specification should be evaluated for accuracy, completeness, consistency, testability, correctness, and clarity. For example, software requirements should be evaluated to verify that:

- There are no internal inconsistencies among requirements;
- All of the performance requirements for the system have been spelled out;
- Fault tolerance, safety, and security requirements are complete and correct;
- Allocation of software functions is accurate and complete;
- Software requirements are appropriate for the system hazards; and
- All requirements are expressed in terms that are measurable or objectively verifiable.

A software requirements traceability analysis should be conducted to trace software requirements to (and from) system requirements and to risk analysis results. In addition to any other analyses and documentation used to verify software requirements, a formal design review is recommended to confirm that requirements are fully specified and appropriate before extensive software design efforts begin. Requirements can be approved and released incrementally, but care should be taken that interactions and interfaces among software (and hardware) requirements are properly reviewed, analyzed, and controlled.

Typical Tasks – Requirements

- Preliminary Risk Analysis
- Traceability Analysis
  - Software Requirements to System Requirements (and vice versa)
  - Software Requirements to Risk Analysis
- Description of User Characteristics
- Listing of Characteristics and Limitations of Primary and Secondary Memory
- Software Requirements Evaluation
- Software User Interface Requirements Analysis
- System Test Plan Generation
- Acceptance Test Plan Generation
- Ambiguity Review or Analysis

5.2.3. Design

In the design process, the software requirements specification is translated into a logical and physical representation of the software to be implemented. The software design specification is a description of what the software should do and how it should do it. Due to complexity of the project or to enable persons with varying levels of technical responsibilities to clearly understand design information, the design specification may contain both a high level summary of the design and detailed design information. The completed software design specification constrains the programmer/coder to stay within the intent of the agreed upon requirements and design. A complete software design specification will relieve the programmer from the need to make ad hoc design decisions.

The software design needs to address human factors. Use error caused by designs that are either overly complex or contrary to users' intuitive expectations for operation is one of the most persistent and critical problems encountered by FDA. Frequently, the design of the software is a factor in such use errors. Human factors engineering should be woven into the entire design and development process, including the device design requirements, analyses, and tests. Device safety and usability issues should be considered when developing flowcharts, state diagrams, prototyping tools, and test plans. Also, task and function analyses, risk analyses, prototype tests and reviews, and full usability tests should be performed. Participants from the user population should be included when applying these methodologies.

The software design specification should include:

- Software requirements specification, including predetermined criteria for acceptance of the software;
- Software risk analysis;
- Development procedures and coding guidelines (or other programming procedures);
- Systems documentation (e.g., a narrative or a context diagram) that describes the systems context in which the program is intended to function, including the relationship of hardware, software, and the physical environment;
- Hardware to be used;
- Parameters to be measured or recorded;
- Logical structure (including control logic) and logical processing steps (e.g., algorithms);
- Data structures and data flow diagrams;
- Definitions of variables (control and data) and description of where they are used;
- Error, alarm, and warning messages;
- Supporting software (e.g., operating systems, drivers, other application software);
- Communication links (links among internal modules of the software, links with the supporting software, links with the hardware, and links with the user);
- Security measures (both physical and logical security); and
- Any additional constraints not identified in the above elements.

The first four of the elements noted above usually are separate pre-existing documents that are included by reference in the software design specification. Software requirements specification was discussed in the preceding section, as was software risk analysis. Written development procedures serve as a guide to the organization, and written programming procedures serve as a guide to individual programmers. As software cannot be validated without knowledge of the context in which it is intended to function, systems documentation is referenced. If some of the above elements are not included in the software, it may be helpful to future reviewers and maintainers of the software if that is clearly stated (e.g., There are no error messages in this program).

The activities that occur during software design have several purposes. Software design evaluations are conducted to determine if the design is complete, correct, consistent, unambiguous, feasible, and maintainable. Appropriate consideration of software architecture (e.g., modular structure) during design can reduce the magnitude of future validation efforts when software changes are needed. Software design evaluations may include analyses of control flow, data flow, complexity, timing, sizing, memory allocation, criticality analysis, and many other aspects of the design. A traceability analysis should be conducted to verify that the software design implements all of the software requirements. As a technique for identifying where requirements are not sufficient, the traceability analysis should also verify that all aspects of the design are traceable to software requirements. An analysis of communication links should be conducted to evaluate the proposed design with respect to hardware, user, and related software requirements. The software risk analysis should be re-examined to determine whether any additional hazards have been identified and whether any new hazards have been introduced by the design.

At the end of the software design activity, a Formal Design Review should be conducted to verify that the design is correct, consistent, complete, accurate, and testable, before moving to implement the design. Portions of the design can be approved and released incrementally for implementation; but care should be taken that interactions and communication links among various elements are properly reviewed, analyzed, and controlled.

Most software development models will be iterative. This is likely to result in several versions of both the software requirement specification and the software design specification. All approved versions should be archived and controlled in accordance with established configuration management procedures.

Typical Tasks – Design

- Updated Software Risk Analysis
- Traceability Analysis - Design Specification to Software Requirements (and vice versa)
- Software Design Evaluation
- Design Communication Link Analysis
- Module Test Plan Generation
- Integration Test Plan Generation
- Test Design Generation (module, integration, system, and acceptance)

5.2.4. Construction or Coding

Software may be constructed either by coding (i.e., programming) or by assembling together previously coded software components (e.g., from code libraries, off-the-shelf software, etc.) for use in a new application. Coding is the software activity where the detailed design specification is implemented as source code. Coding is the lowest level of abstraction for the software development process. It is the last stage in decomposition of the software requirements where module specifications are translated into a programming language.

Coding usually involves the use of a high-level programming language, but may also entail the use of assembly language (or microcode) for time-critical operations. The source code may be either compiled or interpreted for use on a target hardware platform. Decisions on the selection of programming languages and software build tools (assemblers, linkers, and compilers) should include consideration of the impact on subsequent quality evaluation tasks (e.g., availability of debugging and testing tools for the chosen language). Some compilers offer optional levels and commands for error checking to assist in debugging the code. Different levels of error checking may be used throughout the coding process, and warnings or other messages from the compiler may or may not be recorded. However, at the end of the coding and debugging process, the most rigorous level of error checking is normally used to document what compilation errors still remain in the software. If the most rigorous level of error checking is not used for final translation of the source code, then justification for use of the less rigorous translation error checking should be documented. Also, for the final compilation, there should be documentation of the compilation process and its outcome, including any warnings or other messages from the compiler and their resolution, or justification for the decision to leave issues unresolved.

Firms frequently adopt specific coding guidelines that establish quality policies and procedures related to the software coding process. Source code should be evaluated to verify its compliance with specified coding guidelines. Such guidelines should include coding conventions regarding clarity, style, complexity management, and commenting. Code comments should provide useful and descriptive information for a module, including expected inputs and outputs, variables referenced, expected data types, and operations to be performed. Source code should also be evaluated to verify its compliance with the corresponding detailed design specification. Modules ready for integration and test should have documentation of compliance with coding guidelines and any other applicable quality policies and procedures.

Source code evaluations are often implemented as code inspections and code walkthroughs. Such static analyses provide a very effective means to detect errors before execution of the code. They allow for examination of each error in isolation and can also help in focusing later dynamic testing of the software. Firms may use manual (desk) checking with appropriate controls to ensure consistency and independence. Source code evaluations should be extended to verification of internal linkages between modules and layers (horizontal and vertical interfaces), and compliance with their design specifications. Documentation of the procedures used and the results of source code evaluations should be maintained as part of design verification.

A source code traceability analysis is an important tool to verify that all code is linked to established specifications and established test procedures. A source code traceability analysis should be conducted and documented to verify that:

- Each element of the software design specification has been implemented in code;
- Modules and functions implemented in code can be traced back to an element in the software design specification and to the risk analysis;
- Tests for modules and functions can be traced back to an element in the software design specification and to the risk analysis; and
- Tests for modules and functions can be traced to source code for the same modules and functions.

<u>Typical Tasks – Construction or Coding</u>

- Traceability Analyses
  - Source Code to Design Specification (and vice versa)
  - Test Cases to Source Code and to Design Specification
- Source Code and Source Code Documentation Evaluation
- Source Code Interface Analysis
- Test Procedure and Test Case Generation (module, integration, system, and acceptance)

5.2.5. Testing by the Software Developer

Software testing entails running software products under known conditions with defined inputs and documented outcomes that can be compared to their predefined expectations. It is a time consuming, difficult, and imperfect activity. As such, it requires early planning in order to be effective and efficient.

Test plans and test cases should be created as early in the software development process as feasible. They should identify the schedules, environments, resources (personnel, tools, etc.), methodologies, cases (inputs, procedures, outputs, expected results), documentation, and reporting criteria. The magnitude of effort to be applied throughout the testing process can be linked to complexity, criticality, reliability, and/or safety issues (e.g., requiring functions or modules that produce critical outcomes to be challenged with intensive testing of their fault tolerance features). Descriptions of categories of software and software testing effort appear in the literature, for example:

- NIST Special Publication 500-235, *Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric*;
- NUREG/CR-6293, *Verification and Validation Guidelines for High Integrity Systems*; and
- IEEE Computer Society Press, *Handbook of Software Reliability Engineering*.

Software test plans should identify the particular tasks to be conducted at each stage of development and include justification of the level of effort represented by their corresponding completion criteria.

Software testing has limitations that must be recognized and considered when planning the testing of a particular software product. Except for the simplest of programs, software cannot be exhaustively tested. Generally it is not feasible to test a software product with all possible inputs, nor is it possible to test all possible data processing paths that can occur during program execution. There is no one type of testing or testing methodology that can ensure a particular software product has been thoroughly tested. Testing of all program functionality does not mean all of the program has been tested. Testing of all of a program's code does not mean all necessary functionality is present in the program. Testing of all program functionality and all program code does not mean the program is 100% correct! Software testing that finds no errors should not be interpreted to mean that errors do not exist in the software product; it may mean the testing was superficial.

An essential element of a software test case is the expected result. It is the key detail that permits objective evaluation of the actual test result. This necessary testing information is obtained from the corresponding, predefined definition or specification. A software specification document must identify what, when, how, why, etc., is to be achieved with an engineering (i.e., measurable or objectively verifiable) level of detail in order for it to be confirmed through testing. The real effort of effective software testing lies in the definition of what is to be tested rather than in the performance of the test.

A software testing process should be based on principles that foster effective examinations of a software product. Applicable software testing tenets include:

- The expected test outcome is predefined;
- A good test case has a high probability of exposing an error;
- A successful test is one that finds an error;
- There is independence from coding;
- Both application (user) and software (programming) expertise are employed;
- Testers use different tools from coders;
- Examining only the usual case is insufficient;
- Test documentation permits its reuse and an independent confirmation of the pass/fail status of a test outcome during subsequent review.

Once the prerequisite tasks (e.g., code inspection) have been successfully completed, software testing begins. It starts with unit level testing and concludes with system level testing. There may be a distinct integration level of testing. A software product should be challenged with test cases based on its internal structure and with test cases based on its external specification. These tests should provide a thorough and rigorous examination of the software product's compliance with its functional, performance, and interface definitions and requirements.

Code-based testing is also known as structural testing or "white-box" testing. It identifies test cases based on knowledge obtained from the source code, detailed design specification, and other development documents. These test cases challenge the control decisions made by the program; and the program's data structures including configuration tables. Structural testing can identify "dead" code that is never executed when the program is run. Structural testing is accomplished primarily with unit (module) level testing, but can be extended to other levels of software testing.

The level of structural testing can be evaluated using metrics that are designed to show what percentage of the software structure has been evaluated during structural testing. These metrics are typically referred to as “coverage” and are a measure of completeness with respect to test selection criteria. The amount of structural coverage should be commensurate with the level of risk posed by the software. Use of the term “coverage” usually means 100% coverage. For example, if a testing program has achieved “statement coverage,” it means that 100% of the statements in the software have been executed at least once. Common structural coverage metrics include:

- **Statement Coverage** – This criteria requires sufficient test cases for each program statement to be executed at least once; however, its achievement is insufficient to provide confidence in a software product's behavior.

- **Decision (Branch) Coverage** – This criteria requires sufficient test cases for each program decision or branch to be executed so that each possible outcome occurs at least once. It is considered to be a minimum level of coverage for most software products, but decision coverage alone is insufficient for high-integrity applications.

- **Condition Coverage** – This criteria requires sufficient test cases for each condition in a program decision to take on all possible outcomes at least once. It differs from branch coverage only when multiple conditions must be evaluated to reach a decision.

- **Multi-Condition Coverage** – This criteria requires sufficient test cases to exercise all possible combinations of conditions in a program decision.

- **Loop Coverage** – This criteria requires sufficient test cases for all program loops to be executed for zero, one, two, and many iterations covering initialization, typical running and termination (boundary) conditions.

- **Path Coverage** – This criteria requires sufficient test cases for each feasible path, basis path, etc., from start to exit of a defined program segment, to be executed at least once. Because of the very large number of possible paths through a software program, path coverage is generally not achievable. The amount of path coverage is normally established based on the risk or criticality of the software under test.

- **Data Flow Coverage** – This criteria requires sufficient test cases for each feasible data flow to be executed at least once. A number of data flow testing strategies are available.

Definition-based or specification-based testing is also known as functional testing or "black-box" testing. It identifies test cases based on the definition of what the software product (whether it be a unit (module) or a complete program) is intended to do. These test cases challenge the intended use or functionality of a program, and the program's internal and external interfaces. Functional testing can be applied at all levels of software testing, from unit to system level testing.

The following types of functional software testing involve generally increasing levels of effort:

- **Normal Case –** Testing with usual inputs is necessary. However, testing a software product only with expected, valid inputs does not thoroughly test that software product. By itself, normal case testing cannot provide sufficient confidence in the dependability of the software product.

- **Output Forcing –** Choosing test inputs to ensure that selected (or all) software outputs are generated by testing.

- **Robustness –** Software testing should demonstrate that a software product behaves correctly when given unexpected, invalid inputs. Methods for identifying a sufficient set of such test cases include Equivalence Class Partitioning, Boundary Value Analysis, and Special Case Identification (Error Guessing). While important and necessary, these techniques do not ensure that all of the most appropriate challenges to a software product have been identified for testing.

- **Combinations of Inputs –** The functional testing methods identified above all emphasize individual or single test inputs. Most software products operate with multiple inputs under their conditions of use. Thorough software product testing should consider the combinations of inputs a software unit or system may encounter during operation. Error guessing can be extended to identify combinations of inputs, but it is an ad hoc technique. Cause-effect graphing is one functional software testing technique that systematically identifies combinations of inputs to a software product for inclusion in test cases.

Functional and structural software test case identification techniques provide specific inputs for testing, rather than random test inputs. One weakness of these techniques is the difficulty in linking structural and functional test completion criteria to a software product's reliability. Advanced software testing methods, such as statistical testing, can be employed to provide further assurance that a software product is dependable. Statistical testing uses randomly generated test data from defined distributions based on an operational profile (e.g., expected use, hazardous use, or malicious use of the software product). Large amounts of test data are generated and can be targeted to cover particular areas or concerns, providing an increased possibility of identifying individual and multiple rare operating conditions that were not anticipated by either the software product's designers or its testers. Statistical testing also provides high structural coverage. It does require a stable software product. Thus, structural and functional testing are prerequisites for statistical testing of a software product.

Another aspect of software testing is the testing of software changes. Changes occur frequently during software development. These changes are the result of 1) debugging that finds an error and it is corrected, 2) new or changed requirements ("requirements creep"), and 3) modified designs as more effective or efficient implementations are found. Once a software product has been baselined (approved), any change to that product should have its own “mini life cycle,” including testing. Testing of a changed software product requires additional effort. Not only should it demonstrate that the change was implemented correctly, testing should also demonstrate that the change did not adversely impact other parts of the software product. Regression analysis and testing are employed to provide assurance that a change has not created problems elsewhere in the software product. Regression analysis is the determination of the impact of a change based on review of the relevant documentation (e.g., software requirements specification, software design specification, source code, test plans, test cases, test scripts, etc.) in order to identify the necessary regression tests to be run. Regression testing is the rerunning of test cases that a program has previously executed correctly and comparing the current result to the previous result in order to detect unintended effects of a software change. Regression analysis and regression testing should also be employed when using integration methods to build a software product to ensure that newly integrated modules do not adversely impact the operation of previously integrated modules.

In order to provide a thorough and rigorous examination of a software product, development testing is typically organized into levels. As an example, a software product's testing can be organized into unit, integration, and system levels of testing.

1) Unit (module or component) level testing focuses on the early examination of sub-program functionality and ensures that functionality not visible at the system level is examined by testing. Unit testing ensures that quality software units are furnished for integration into the finished software product.

2) Integration level testing focuses on the transfer of data and control across a program's internal and external interfaces. External interfaces are those with other software (including operating system software), system hardware, and the users and can be described as communications links.

3) System level testing demonstrates that all specified functionality exists and that the software product is trustworthy. This testing verifies the as-built program's functionality and performance with respect to the requirements for the software product as exhibited on the specified operating platform(s). System level software testing addresses functional concerns and the following elements of a device's software that are related to the intended use(s):

   - Performance issues (e.g., response times, reliability measurements);
   - Responses to stress conditions, e.g., behavior under maximum load, continuous use;
   - Operation of internal and external security features;
   - Effectiveness of recovery procedures, including disaster recovery;
   - Usability;
   - Compatibility with other software products;
   - Behavior in each of the defined hardware configurations; and
   - Accuracy of documentation.

Control measures (e.g., a traceability analysis) should be used to ensure that the intended coverage is achieved.

System level testing also exhibits the software product's behavior in the intended operating environment. The location of such testing is dependent upon the software developer's ability to produce the target operating environment(s). Depending upon the circumstances, simulation and/or testing at (potential) customer locations may be utilized. Test plans should identify the controls needed to ensure that the intended coverage is achieved and that proper documentation is prepared when planned system level testing is conducted at sites not directly controlled by the software developer. Also, for a software product that is a medical device or a component of a medical device that is to be used on humans prior to FDA clearance, testing involving human subjects may require an Investigational Device Exemption (IDE) or Institutional Review Board (IRB) approval.

Test procedures, test data, and test results should be documented in a manner permitting objective pass/fail decisions to be reached. They should also be suitable for review and objective decision making subsequent to running the test, and they should be suitable for use in any subsequent regression testing. Errors detected during testing should be logged, classified, reviewed, and resolved prior to release of the software. Software error data that is collected and analyzed during a development life cycle may be used to determine the suitability of the software product for release for commercial distribution. Test reports should comply with the requirements of the corresponding test plans.

Software products that perform useful functions in medical devices or their production are often complex. Software testing tools are frequently used to ensure consistency, thoroughness, and efficiency in the testing of such software products and to fulfill the requirements of the planned testing activities. These tools may include supporting software built in-house to facilitate unit (module) testing and subsequent integration testing (e.g., drivers and stubs) as well as commercial software testing tools. Such tools should have a degree of quality no less than the software product they are used to develop. Appropriate documentation providing evidence of the validation of these software tools for their intended use should be maintained (see section 6 of this guidance).

<u>Typical Tasks – Testing by the Software Developer</u>

- Test Planning
- Structural Test Case Identification
- Functional Test Case Identification
- Traceability Analysis - Testing
  - Unit (Module) Tests to Detailed Design
  - Integration Tests to High Level Design
  - System Tests to Software Requirements
- Unit (Module) Test Execution
- Integration Test Execution
- Functional Test Execution
- System Test Execution
- Acceptance Test Execution
- Test Results Evaluation
- Error Evaluation/Resolution
- Final Test Report

5.2.6. User Site Testing

Testing at the user site is an essential part of software validation. The Quality System regulation requires installation and inspection procedures (including testing where appropriate) as well as documentation of inspection and testing to demonstrate proper installation. (See 21 CFR §820.170.) Likewise, manufacturing equipment must meet specified requirements, and automated systems must be validated for their intended use. (See 21 CFR §820.70(g) and 21 CFR §820.70(i) respectively.)

Terminology regarding user site testing can be confusing. Terms such as beta test, site validation, user acceptance test, installation verification, and installation testing have all been used to describe user site testing. For purposes of this guidance, the term "user site testing" encompasses all of these and any other testing that takes place outside of the developer's controlled environment. This testing should take place at a user's site with the actual hardware and software that will be part of the installed system configuration. The testing is accomplished through either actual or simulated use of the software being tested within the context in which it is intended to function.

Guidance contained here is general in nature and is applicable to any user site testing. However, in some areas (e.g., blood establishment systems) there may be specific site validation issues that need to be considered in the planning of user site testing. Test planners should check with the FDA Center(s) with the corresponding product jurisdiction to determine whether there are any additional regulatory requirements for user site testing.

User site testing should follow a pre-defined written plan with a formal summary of testing and a record of formal acceptance. Documented evidence of all testing procedures, test input data, and test results should be retained.

There should be evidence that hardware and software are installed and configured as specified. Measures should ensure that all system components are exercised during the testing and that the versions of these components are those specified. The testing plan should specify testing throughout the full range of operating conditions and should specify continuation for a sufficient time to allow the system to encounter a wide spectrum of conditions and events in an effort to detect any latent faults that are not apparent during more normal activities.

Some of the evaluations that have been performed earlier by the software developer at the developer's site should be repeated at the site of actual use. These may include tests for a high volume of data, heavy loads or stresses, security, fault testing (avoidance, detection, tolerance, and recovery), error messages, and implementation of safety requirements. The developer may be able to furnish the user with some of the test data sets to be used for this purpose.

In addition to an evaluation of the system's ability to properly perform its intended functions, there should be an evaluation of the ability of the users of the system to understand and correctly interface with it. Operators should be able to perform the intended functions and respond in an appropriate and timely manner to all alarms, warnings, and error messages.

During user site testing, records should be maintained of both proper system performance and any system failures that are encountered. The revision of the system to compensate for faults detected during this user site testing should follow the same procedures and controls as for any other software change.

The developers of the software may or may not be involved in the user site testing. If the developers are involved, they may seamlessly carry over to the user's site the last portions of design-level systems testing. If the developers are not involved, it is all the more important that the user have persons who understand the importance of careful test planning, the definition of expected test results, and the recording of all test outputs.

Typical Tasks – User Site Testing

- Acceptance Test Execution
- Test Results Evaluation
- Error Evaluation/Resolution
- Final Test Report

5.2.7. Maintenance and Software Changes

As applied to software, the term maintenance does not mean the same as when applied to hardware. The operational maintenance of hardware and software are different because their failure/error mechanisms are different. Hardware maintenance typically includes preventive hardware maintenance actions, component replacement, and corrective changes. Software maintenance includes corrective, perfective, and adaptive maintenance but does not include preventive maintenance actions or software component replacement.

Changes made to correct errors and faults in the software are corrective maintenance. Changes made to the software to improve the performance, maintainability, or other attributes of the software system are perfective maintenance. Software changes to make the software system usable in a changed environment are adaptive maintenance.

When changes are made to a software system, either during initial development or during post release maintenance, sufficient regression analysis and testing should be conducted to demonstrate that portions of the software not involved in the change were not adversely impacted. This is in addition to testing that evaluates the correctness of the implemented change(s).

The specific validation effort necessary for each software change is determined by the type of change, the development products affected, and the impact of those products on the operation of the software. Careful and complete documentation of the design structure and interrelationships of various modules, interfaces, etc., can limit the validation effort needed when a change is made. The level of effort needed to fully validate a change is also dependent upon the degree to which validation of the original software was documented and archived. For example, test documentation, test cases, and results of previous verification and validation testing need to be archived if they are to be available for performing subsequent regression testing. Failure to archive this information for later use can significantly increase the level of effort and expense of revalidating the software after a change is made.

In addition to software verification and validation tasks that are part of the standard software development process, the following additional maintenance tasks should be addressed:

- **Software Validation Plan Revision** - For software that was previously validated, the existing software validation plan should be revised to support the validation of the revised software. If no previous software validation plan exists, such a plan should be established to support the validation of the revised software.

- **Anomaly Evaluation** – Software organizations frequently maintain documentation, such as software problem reports that describe software anomalies discovered and the specific corrective action taken to fix each anomaly. Too often, however, mistakes are repeated because software developers do not take the next step to determine the root causes of problems and make the process and procedural changes needed to avoid recurrence of the problem. Software anomalies should be evaluated in terms of their severity and their effects on system operation and safety, but they should also be treated as symptoms of process deficiencies in the quality system. A root cause analysis of anomalies can identify specific quality system deficiencies. Where trends are identified (e.g., recurrence of similar software anomalies), appropriate corrective and preventive actions must be implemented and documented to avoid further recurrence of similar quality problems. (See 21 CFR 820.100.)

- **Problem Identification and Resolution Tracking** - All problems discovered during maintenance of the software should be documented. The resolution of each problem should be tracked to ensure it is fixed, for historical reference, and for trending.

- **Proposed Change Assessment** - All proposed modifications, enhancements, or additions should be assessed to determine the effect each change would have on the system. This information should determine the extent to which verification and/or validation tasks need to be iterated.

- **Task Iteration** - For approved software changes, all necessary verification and validation tasks should be performed to ensure that planned changes are implemented correctly, all documentation is complete and up to date, and no unacceptable changes have occurred in software performance.

- **Documentation Updating** – Documentation should be carefully reviewed to determine which documents have been impacted by a change. All approved documents (e.g., specifications, test procedures, user manuals, etc.) that have been affected should be updated in accordance with configuration management procedures. Specifications should be updated before any maintenance and software changes are made.

SECTION 6. VALIDATION OF AUTOMATED PROCESS EQUIPMENT AND QUALITY SYSTEM SOFTWARE

The Quality System regulation requires that "when computers or automated data processing systems are used as part of production or the quality system, the [device] manufacturer shall validate computer software for its intended use according to an established protocol." (See 21 CFR §820.70(i)). This has been a regulatory requirement of FDA's medical device Good Manufacturing Practice (GMP) regulations since 1978.

In addition to the above validation requirement, computer systems that implement part of a device manufacturer's production processes or quality system (or that are used to create and maintain records required by any other FDA regulation) are subject to the Electronic Records; Electronic Signatures regulation. (See 21 CFR Part 11.) This regulation establishes additional security, data integrity, and validation requirements when records are created or maintained electronically. These additional Part 11 requirements should be carefully considered and included in system requirements and software requirements for any automated record `keeping systems. System validation and software validation should demonstrate that all Part 11 requirements have been met.

Computers and automated equipment are used extensively throughout all aspects of medical device design, laboratory testing and analysis, product inspection and acceptance, production and process control, environmental controls, packaging, labeling, traceability, document control, complaint management, and many other aspects of the quality system. Increasingly, automated plant floor operations can involve extensive use of embedded systems in:

- programmable logic controllers;
- digital function controllers;
- statistical process control;
- supervisory control and data acquisition;
- robotics;
- human-machine interfaces;
- input/output devices; and
- computer operating systems.

Software tools are frequently used to design, build, and test the software that goes into an automated medical device. Many other commercial software applications, such as word processors, spreadsheets, databases, and flowcharting software are used to implement the quality system. All of these applications are subject to the requirement for software validation, but the validation approach used for each application can vary widely.

Whether production or quality system software is developed in-house by the device manufacturer, developed by a contractor, or purchased off-the-shelf, it should be developed using the basic principles outlined elsewhere in this guidance. The device manufacturer has latitude and flexibility in defining how validation of that software will be accomplished, but validation should be a key consideration in deciding how and by whom the software will be developed or from whom it will be purchased. The software developer defines a life cycle model. Validation is typically supported by:

- verifications of the outputs from each stage of that software development life cycle; and
- checking for proper operation of the finished software in the device manufacturer's intended use environment.

6.1. HOW MUCH VALIDATION EVIDENCE IS NEEDED?

The level of validation effort should be commensurate with the risk posed by the automated operation. In addition to risk other factors, such as the complexity of the process software and the degree to which the device manufacturer is dependent upon that automated process to produce a safe and effective device, determine the nature and extent of testing needed as part of the validation effort. Documented requirements and risk analysis of the automated process help to define the scope of the evidence needed to show that the software is validated for its intended use. For example, an automated milling machine may require very little testing if the device manufacturer can show that the output of the operation is subsequently fully verified against the specification before release. On the other hand, extensive testing may be needed for:

- a plant-wide electronic record and electronic signature system;
- an automated controller for a sterilization cycle; or
- automated test equipment used for inspection and acceptance of finished circuit boards in a life-sustaining / life-supporting device.

Numerous commercial software applications may be used as part of the quality system (e.g., a spreadsheet or statistical package used for quality system calculations, a graphics package used for trend analysis, or a commercial database used for recording device history records or for complaint management). The extent of validation evidence needed for such software depends on the device manufacturer's documented intended use of that software. For example, a device manufacturer who chooses not to use all the vendor-supplied capabilities of the software only needs to validate those functions that will be used and for which the device manufacturer is dependent upon the software results as part of production or the quality system. However, high risk applications should not be running in the same operating environment with non-validated software functions, even if those software functions are not used. Risk mitigation techniques such as memory partitioning or other approaches to resource protection may need to be considered when high risk applications and lower risk applications are to be used in the same operating environment. When software is upgraded or any changes are made to the software, the device manufacturer should consider how those changes may impact the "used portions" of the software and must reconfirm the validation of those portions of the software that are used. (See 21 CFR §820.70(i).)

6.2. DEFINED USER REQUIREMENTS

A very important key to software validation is a documented user requirements specification that defines:

- the “intended use” of the software or automated equipment; and
- the extent to which the device manufacturer is dependent upon that software or equipment for production of a quality medical device.

The device manufacturer (user) needs to define the expected operating environment including any required hardware and software configurations, software versions, utilities, etc. The user also needs to:

- document requirements for system performance, quality, error handling, startup, shutdown, security, etc.;
- identify any safety related functions or features, such as sensors, alarms, interlocks, logical processing steps, or command sequences; and
- define objective criteria for determining acceptable performance.

The validation must be conducted in accordance with a documented protocol, and the validation results must also be documented. (See 21 CFR §820.70(i).) Test cases should be documented that will exercise the system to challenge its performance against the pre-determined criteria, especially for its most critical parameters. Test cases should address error and alarm conditions, startup, shutdown, all applicable user functions and operator controls, potential operator errors, maximum and minimum ranges of allowed values, and stress conditions applicable to the intended use of the equipment. The test cases should be executed and the results should be recorded and evaluated to determine whether the results support a conclusion that the software is validated for its intended use.

A device manufacturer may conduct a validation using their own personnel or may depend on a third party such as the equipment/software vendor or a consultant. In any case, the device manufacturer retains the ultimate responsibility for ensuring that the production and quality system software:

- is validated according to a written procedure for the particular intended use; and
- will perform as intended in the chosen application.

The device manufacturer should have documentation including:

- defined user requirements;
- validation protocol used;
- acceptance criteria;
- test cases and results; and
- a validation summary

that objectively confirms that the software is validated for its intended use.

6.3. VALIDATION OF OFF-THE-SHELF SOFTWARE AND AUTOMATED EQUIPMENT

Most of the automated equipment and systems used by device manufacturers are supplied by third-party vendors and are purchased off-the-shelf (OTS). The device manufacturer is responsible for ensuring that the product development methodologies used by the OTS software developer are appropriate and sufficient for the device manufacturer's intended use of that OTS software. For OTS software and equipment, the device manufacturer may or may not have access to the vendor's software validation documentation. If the vendor can provide information about their system requirements, software requirements, validation process, and the results of their validation, the medical device manufacturer can use that information as a beginning point for their required validation documentation. The vendor's life cycle documentation, such as testing protocols and results, source code, design specification, and requirements specification, can be useful in establishing that the software has been validated. However, such documentation is frequently not available from commercial equipment vendors, or the vendor may refuse to share their proprietary information.

Where possible and depending upon the device risk involved, the device manufacturer should consider auditing the vendor's design and development methodologies used in the construction of the OTS software and should assess the development and validation documentation generated for the OTS software. Such audits can be conducted by the device manufacturer or by a qualified third party. The audit should demonstrate that the vendor's procedures for and results of the verification and validation activities performed the OTS software are appropriate and sufficient for the safety and effectiveness requirements of the medical device to be produced using that software.

Some vendors who are not accustomed to operating in a regulated environment may not have a documented life cycle process that can support the device manufacturer's validation requirement. Other vendors may not permit an audit. Where necessary validation information is not available from the vendor, the device manufacturer will need to perform sufficient system level "black box" testing to establish that the software meets their "user needs and intended uses." For many applications black box testing alone is not sufficient. Depending upon the risk of the device produced, the role of the OTS software in the process, the ability to audit the vendor, and the sufficiency of vendor-supplied information, the use of OTS software or equipment may or may not be appropriate, especially if there are suitable alternatives available. The device manufacturer should also consider the implications (if any) for continued maintenance and support of the OTS software should the vendor terminate their support.

For some off-the-shelf software development tools, such as software compilers, linkers, editors, and operating systems, exhaustive black-box testing by the device manufacturer may be impractical. Without such testing – a key element of the validation effort – it may not be possible to validate these software tools. However, their proper operation may be satisfactorily inferred by other means. For example, compilers are frequently certified by independent third-party testing, and commercial software products may have "bug lists", system requirements and other operational information available from the vendor that can be compared to the device manufacturer's intended use to help focus the "black-box" testing effort. Off-the-shelf operating systems need not be validated as a separate program. However, system-level validation testing of the application software should address all the operating system services used, including maximum loading conditions, file operations, handling of system error conditions, and memory constraints that may be applicable to the intended use of the application program.

For more detailed information, see the production and process software references in Appendix A.

APPENDIX A - REFERENCES

Food and Drug Administration References

Design Control Guidance for Medical Device Manufacturers, Center for Devices and Radiological Health, Food and Drug Administration, March 1997.

Do It by Design, An Introduction to Human Factors in Medical Devices, Center for Devices and Radiological Health, Food and Drug Administration, March 1997.

Electronic Records; Electronic Signatures Final Rule, 62 Federal Register 13430 (March 20, 1997).

Glossary of Computerized System and Software Development Terminology, Division of Field Investigations, Office of Regional Operations, Office of Regulatory Affairs, Food and Drug Administration, August 1995.

Guidance for the Content of Pre-market Submissions for Software Contained in Medical Devices, Office of Device Evaluation, Center for Devices and Radiological Health, Food and Drug Administration, May 1998.

Guidance for Industry, FDA Reviewers and Compliance on Off-the-Shelf Software Use in Medical Devices, Office of Device Evaluation, Center for Devices and Radiological Health, Food and Drug Administration, September 1999.

Guideline on General Principles of Process Validation, Center for Drugs and Biologics, & Center For Devices and Radiological Health, Food and Drug Administration, May 1987.

Medical Devices; Current Good Manufacturing Practice (CGMP) Final Rule; Quality System Regulation , 61 Federal Register 52602 (October 7, 1996).

Reviewer Guidance for a Pre-Market Notification Submission for Blood Establishment Computer Software, Center for Biologics Evaluation and Research, Food and Drug Administration, January 1997

Student Manual 1, Course INV545, Computer System Validation, Division of Human Resource Development, Office of Regulatory Affairs, Food and Drug Administration, 1997.

Technical Report, Software Development Activities, Division of Field Investigations, Office of Regional Operations, Office of Regulatory Affairs, Food and Drug Administration, July 1987.

Other Government References

W. Richards Adrion, Martha A. Branstad, John C. Cherniavsky. *NBS Special Publication 500-75, Validation, Verification, and Testing of Computer Software,* Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, February 1981.

Martha A. Branstad, John C Cherniavsky, W. Richards Adrion, *NBS Special Publication 500-56, Validation, Verification, and Testing for the Individual Programmer*, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, February 1980.

J.L. Bryant, N.P. Wilburn, *Handbook of Software Quality Assurance Techniques Applicable to the Nuclear Industry*, NUREG/CR-4640, U.S. Nuclear Regulatory Commission, 1987.

H. Hecht, et.al., *Verification and Validation Guidelines for High Integrity Systems.* NUREG/CR-6293. Prepared for U.S. Nuclear Regulatory Commission, 1995.

H. Hecht, et.al., *Review Guidelines on Software Languages for Use in Nuclear Power Plant Safety Systems, Final Report*. NUREG/CR-6463. Prepared for U.S. Nuclear Regulatory Commission, 1996.

J.D. Lawrence, W.L. Persons, *Survey of Industry Methods for Producing Highly Reliable Software*, NUREG/CR-6278, U.S. Nuclear Regulatory Commission, 1994.

J.D. Lawrence, G.G. Preckshot, *Design Factors for Safety-Critical Software,* NUREG/CR-6294, U.S. Nuclear Regulatory Commission, 1994.

Patricia B. Powell, Editor. *NBS Special Publication 500-98, Planning for Software Validation, Verification, and Testing,* Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, November 1982.

Patricia B. Powell, Editor. *NBS Special Publication 500-93, Software Validation, Verification, and Testing Technique and Tool Reference Guide*, Center for Programming Science and Technology, Institute for Computer Sciences and Technology, National Bureau of Standards, U.S. Department of Commerce, September 1982.

Delores R. Wallace, Roger U. Fujii, *NIST Special Publication 500-165, Software Verification and Validation: Its Role in Computer Assurance and Its Relationship with Software Project Management Standards*, National Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, September 1995.

Delores R. Wallace, Laura M. Ippolito, D. Richard Kuhn, *NIST Special Publication 500-204, High Integrity Software, Standards and Guidelines*, Computer Systems Laboratory, National Institute of

Standards and Technology, U.S. Department of Commerce, September 1992.

Delores R. Wallace, et.al. *NIST Special Publication 500-234, Reference Information for the Software Verification and Validation Process*. Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, March 1996.

Delores R. Wallace, Editor. *NIST Special Publication 500-235, Structured Testing: A Testing Methodology Using the Cyclomatic Complexity Metric*. Computer Systems Laboratory, National Institute of Standards and Technology, U.S. Department of Commerce, August 1996.

International and National Consensus Standards

ANSI / ANS-10.4-1987, *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs for the Nuclear Industry*, American National Standards Institute, 1987.

ANSI / ASQC Standard D1160-1995, *Formal Design Reviews*, American Society for Quality Control, 1995.

ANSI / UL 1998:1998, *Standard for Safety for Software in Programmable Components,* Underwriters Laboratories, Inc., 1998.

AS 3563.1-1991, *Software Quality Management System, Part 1: Requirements.* Published by Standards Australia [Standards Association of Australia], 1 The Crescent, Homebush, NSW 2140.

AS 3563.2-1991, *Software Quality Management System, Part 2: Implementation Guide*. Published by Standards Australia [Standards Association of Australia], 1 The Crescent, Homebush, NSW 2140.

IEC 60601-1-4:1996, *Medical electrical equipment, Part 1: General requirements for safety, 4. Collateral Standard: Programmable electrical medical systems.* International Electrotechnical Commission, 1996.

IEC 61506:1997, *Industrial process measurement and control – Documentation of application software*. International Electrotechnical Commission, 1997.

IEC 61508:1998, *Functional safety of electrical/electronic/programmable electronic safety-related systems*. International Electrotechnical Commission, 1998.

IEEE Std 1012-1986, *Software Verification and Validation Plans*, Institute for Electrical and Electronics Engineers, 1986.

IEEE Standards Collection, Software Engineering, Institute of Electrical and Electronics Engineers, Inc., 1994. ISBN 1-55937-442-X.

ISO 8402:1994, *Quality management and quality assurance – Vocabulary*. International Organization for Standardization, 1994.

ISO 9000-3:1997, *Quality management and quality assurance standards - Part 3: Guidelines for the application of ISO 9001:1994 to the development, supply, installation and maintenance of computer software*. International Organization for Standardization, 1997.

ISO 9001:1994, *Quality systems – Model for quality assurance in design, development, production, installation, and servicing.* International Organization for Standardization, 1994.

ISO 13485:1996, *Quality systems – Medical devices – Particular requirements for the application of ISO 9001.* International Organization for Standardization, 1996.

ISO/IEC 12119:1994, *Information technology – Software packages – Quality requirements and testing*, Joint Technical Committee ISO/IEC JTC 1, International Organization for Standardization and International Electrotechnical Commission, 1994.

ISO/IEC 12207:1995, *Information technology – Software life cycle processes*, Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 7, International Organization for Standardization and International Electrotechnical Commission, 1995.

ISO/IEC 14598:1999, *Information technology – Software product evaluation*, Joint Technical Committee ISO/IEC JTC 1, Subcommittee SC 7, International Organization for Standardization and International Electrotechnical Commission, 1999.

ISO 14971-1:1998, *Medical Devices – Risk Management – Part 1: Application of Risk Analysis.* International Organization for Standardization, 1998.

Software Considerations in Airborne Systems and Equipment Certification. Special Committee 167 of RTCA. RTCA Inc., Washington, D.C. Tel: 202-833-9339. Document No. RTCA/DO-178B, December 1992.

Production Process Software References

The Application of the Principles of GLP to Computerized Systems, Environmental Monograph #116, Organization for Economic Cooperation and Development (OECD), 1995.

George J. Grigonis, Jr., Edward J. Subak, Jr., and Michael Wyrick, "Validation Key Practices for Computer Systems Used in Regulated Operations," *Pharmaceutical Technology*, June 1997.

Guide to Inspection of Computerized Systems in Drug Processing, Reference Materials and

Training Aids for Investigators, Division of Drug Quality Compliance, Associate Director for Compliance, Office of Drugs, National Center for Drugs and Biologics, & Division of Field Investigations, Associate Director for Field Support, Executive Director of Regional Operations, Food and Drug Administration, February 1983.

Daniel P. Olivier, "Validating Process Software", *FDA Investigator Course: Medical Device Process Validation*, Food and Drug Administration.

GAMP Guide For Validation of Automated Systems in Pharmaceutical Manufacture,Version V3.0, Good Automated Manufacturing Practice (GAMP) Forum, March 1998:
Volume 1, Part 1: User Guide
Part 2: Supplier Guide
Volume 2: Best Practice for User and Suppliers.

Technical Report No. 18, Validation of Computer-Related Systems. PDA Committee on Validation of Computer-Related Systems. PDA Journal of Pharmaceutical Science and Technology, Volume 49, Number 1, January-February 1995 Supplement.

Validation Compliance Annual 1995, International Validation Forum, Inc.

General Software Quality References

Boris Beizer, *Black Box Testing, Techniques for Functional Testing of Software and Systems*, John Wiley & Sons, 1995. ISBN 0-471-12094-4.

Boris Beizer, *Software System Testing and Quality Assurance*, International Thomson Computer Press, 1996. ISBN 1-85032-821-8.

Boris Beizer, *Software Testing Techniques*, Second Edition, Van Nostrand Reinhold, 1990. ISBN 0-442-20672-0.

Richard Bender, *Writing Testable Requirements, Version 1.0*, Bender & Associates, Inc., Larkspur, CA 94777, 1996.

Frederick P. Brooks, Jr., *The Mythical Man-Month, Essays on Software Engineering*, Addison-Wesley Longman, Anniversary Edition, 1995. ISBN 0-201-83595-9.

Silvana Castano, et.al., *Database Security*, ACM Press, Addison-Wesley Publishing Company, 1995. ISBN 0-201-59375-0.

Computerized Data Systems for Nonclinical Safety Assessment, Current Concepts and Quality Assurance, Drug Information Association, Maple Glen, PA, September 1988.

M. S. Deutsch, *Software Verification and Validation, Realistic Project Approaches*, Prentice Hall, 1982.

Robert H. Dunn and Richard S. Ullman, *TQM for Computer Software*, Second Edition, McGraw-Hill, Inc., 1994. ISBN 0-07-018314-7.

Elfriede Dustin, Jeff Rashka, and John Paul, *Automated Software Testing – Introduction, Management and Performance,* Addison Wesley Longman, Inc., 1999. ISBN 0-201-43287-0.

Robert G. Ebenau and Susan H. Strauss, *Software Inspection Process*, McGraw-Hill, 1994. ISBN 0-07-062166-7.

Richard E. Fairley, *Software Engineering Concepts*, McGraw-Hill Publishing Company, 1985. ISBN 0-07-019902-7.

Michael A. Friedman and Jeffrey M. Voas, *Software Assessment - Reliability, Safety, Testability*, Wiley-Interscience, John Wiley & Sons Inc., 1995. ISBN 0-471-01009-X.

Tom Gilb, Dorothy Graham, *Software Inspection*, Addison-Wesley Publishing Company, 1993. ISBN 0-201-63181-4.

Robert B. Grady, *Practical Software Metrics for Project Management and Process Improvement*, PTR Prentice-Hall Inc., 1992. ISBN 0-13-720384-5.

Les Hatton, *Safer C: Developing Software for High-integrity and Safety-critical Systems*, McGraw-Hill Book Company, 1994. ISBN 0-07-707640-0.

Janis V. Halvorsen, *A Software Requirements Specification Document Model for the Medical Device Industry*, Proceedings IEEE SOUTHEASTCON '93, Banking on Technology, April 4th -7th, 1993, Charlotte, North Carolina.

Debra S. Herrmann, *Software Safety and Reliability: Techniques, Approaches and Standards of Key Industrial Sectors,* IEEE Computer Society, 1999. ISBN 0-7695-0299-7.

Bill Hetzel, *The Complete Guide to Software Testing*, Second Edition, A Wiley-QED Publication, John Wiley & Sons, Inc., 1988. ISBN 0-471-56567-9.

Watts S. Humphrey, *A Discipline for Software Engineering*. Addison-Wesley Longman, 1995. ISBN 0-201-54610-8.

Watts S. Humphrey, *Managing the Software Process*, Addison-Wesley Publishing Company, 1989. ISBN 0-201-18095-2.

Capers Jones, *Software Quality, Analysis and Guidelines for Success*, International Thomson Computer Press, 1997. ISBN 1-85032-867-6.

J.M. Juran, Frank M. Gryna, *Quality Planning and Analysis*, Third Edition, , McGraw-Hill, 1993. ISBN 0-07-033183-9.

Stephen H. Kan, *Metrics and Models in Software Quality Engineering*, Addison-Wesley Publishing Company, 1995. ISBN 0-201-63339-6.

Cem Kaner, Jack Falk, Hung Quoc Nguyen, *Testing Computer Software*, Second Edition, Vsn Nostrand Reinhold, 1993. ISBN 0-442-01361-2.

Craig Kaplan, Ralph Clark, Victor Tang, *Secrets of Software Quality, 40 Innovations from IBM*, McGraw-Hill, 1995. ISBN 0-07-911795-3.

Edward Kit, *Software Testing in the Real World*, Addison-Wesley Longman, 1995. ISBN 0-201-87756-2.

Alan Kusinitz, "Software Validation", *Current Issues in Medical Device Quality Systems,* Association for the Advancement of Medical Instrumentation, 1997. ISBN 1-57020-075-0.

Nancy G. Leveson, *Safeware, System Safety and Computers*, Addison-Wesley Publishing Company, 1995. ISBN 0-201-11972-2.

Michael R. Lyu, Editor, *Handbook of Software Reliability Engineering*, IEEE Computer Society Press, McGraw-Hill, 1996. ISBN 0-07-039400-8.

Steven R. Mallory, *Software Development and Quality Assurance for the Healthcare Manufacturing Industries*, Interpharm Press,Inc., 1994. ISBN 0-935184-58-9.

Brian Marick, *The Craft of Software Testing*, Prentice Hall PTR, 1995. ISBN 0-13-177411-5.

Steve McConnell, *Rapid Development*, Microsoft Press, 1996. ISBN 1-55615-900-5.

Glenford J. Myers, *The Art of Software Testing*, John Wiley & Sons, 1979. ISBN 0-471-04328-1.

Peter G. Neumann, *Computer Related Risks*, ACM Press/Addison-Wesley Publishing Co., 1995. ISBN 0-201-55805-X.

Daniel Olivier, *Conducting Software Audits, Auditing Software for Conformance to FDA Requirements*, Computer Application Specialists, San Diego, CA, 1994.

William Perry, *Effective Methods for Software Testing*, John Wiley & Sons, Inc. 1995. ISBN 0-471-06097-6.

William E. Perry, Randall W. Rice, *Surviving the Top Ten Challenges of Software Testing*, Dorset

House Publishing, 1997. ISBN 0-932633-38-2.

Roger S. Pressman, *Software Engineering, A Practitioner's Approach*, Third Edition, McGraw-Hill Inc., 1992. ISBN 0-07-050814-3.

Roger S. Pressman, *A Manager's Guide to Software Engineering*, McGraw-Hill Inc., 1993 ISBN 0-07-050820-8.

A. P. Sage, J. D. Palmer, *Software Systems Engineering*, John Wiley & Sons, 1990.

Joc Sanders, Eugene Curran, *Software Quality*, Addison-Wesley Publishing Co., 1994. ISBN 0-201-63198-9.

Ken Shumate, Marilyn Keller, *Software Specification and Design, A Disciplined Approach for Real-Time Systems*, John Wiley & Sons, 1992. ISBN 0-471-53296-7.

Dennis D. Smith, *Designing Maintainable Software*, Springer-Verlag, 1999. ISBN 0-387-98783-5.

Ian Sommerville, *Software Engineering*, Third Edition, Addison Wesley Publishing Co., 1989. ISBN 0-201-17568-1.

Karl E. Wiegers, *Creating a Software Engineering Culture*, Dorset House Publishing, 1996. ISBN 0-932633-33-1.

Karl E. Wiegers, *Software Inspection, Improving Quality with Software Inspections*, Software Development, April 1995, pages 55-64.

Karl E. Wiegers, *Software Requirements*, Microsoft Press, 1999. ISBN 0-7356-0631-5.

APPENDIX B - DEVELOPMENT TEAM

Center for Devices and Radiological Health

Office of Compliance	Stewart Crumpler
Office of Device Evaluation	James Cheng, Donna-Bea Tillman
Office of Health and Industry Programs	Bryan Benesch, Dick Sawyer
Office of Science and Technology	John Murray
Office of Surveillance and Biometrics	Howard Press

Center for Drug Evaluation and Research

Office of Medical Policy	Charles Snipes

Center for Biologics Evaluation and Research

Office of Compliance and Biologics Quality	Alice Godziemski

Office of Regulatory Affairs

Office of Regional Operations	David Bergeson, Joan Loreng

Guidance for Industry

Computerized Systems Used in Clinical Investigations

U.S. Department of Health and Human Services
Food and Drug Administration (FDA)
Office of the Commissioner (OC)
May 2007

Guidance for Industry

Computerized Systems Used in Clinical Investigations

Additional copies are available from:

Office of Training and Communication
Division of Drug Information
Center for Drug Evaluation and Research (CDER)
(Tel) 301-827-4573
http://www.fda.gov/cder/guidance/index.htm
or
Office of Communication, Training and
Manufacturers Assistance
Center for Biologics Evaluation and Research
http://www.fda.gov/cber/guidelines.htm
(Tel) 800-835-4709 or 301-827-1800
or
Office of Communication, Education, and Radiation Programs
Division of Small Manufacturers, International, and Consumer Assistance
Center for Devices and Radiological Health
http://www.fda.gov/cdrh/ggpmain.html
Email: dsmica@fda.hhs.gov
Fax: 240.276.3151
(Tel) Manufacturers and International Assistance: 800.638.2041 or 240.276.3150
or
Office of Food Additive Safety
Center for Food, Safety and Applied Nutrition
(Tel) 301-436-1200
http://www.cfsan.fda.gov/guidance.html
or
Communications Staff, HFV-12
Center for Veterinary Medicine
(Tel) 240-276-9300
http://www.fda.gov/cvm/guidance/published
or
Good Clinical Practice Programs
Office of the Commissioner

U.S. Department of Health and Human Services
Food and Drug Administration
Office of the Commissioner (OC)
May 2007

TABLE OF CONTENTS

Guidance for Industry[1] Computerized Systems Used in Clinical Investigations

This guidance represents the Food and Drug Administration's (FDA's) current thinking on this topic. It does not create or confer any rights for or on any person and does not operate to bind FDA or the public. You can use an alternative approach if the approach satisfies the requirements of the applicable statutes and regulations. If you want to discuss an alternative approach, contact the FDA staff responsible for implementing this guidance. If you cannot identify the appropriate FDA staff, call the appropriate number listed on the title page of this guidance.

I. INTRODUCTION

This document provides to sponsors, contract research organizations (CROs), data management centers, clinical investigators, and institutional review boards (IRBs), recommendations regarding the use of computerized systems in clinical investigations. The computerized system applies to records in electronic form that are used to create, modify, maintain, archive, retrieve, or transmit clinical data required to be maintained, or submitted to the FDA. Because the source data[2] are necessary for the reconstruction and evaluation of the study to determine the safety of food and color additives and safety and effectiveness of new human and animal drugs,[3] and medical devices, this guidance is intended to assist in ensuring confidence in the reliability, quality, and integrity of electronic source data and source documentation (i.e., electronic records).

This guidance supersedes the guidance of the same name dated April 1999; and supplements the guidance for industry on *Part 11, Electronic Records; Electronic Signatures — Scope and Application* and the Agency's international harmonization efforts[4] when applying these guidances to source data generated at clinical study sites.

[1] This guidance has been prepared by the Office of Critical Path Programs, the Good Clinical Practice Program, and the Office of Regulatory Affairs in cooperation with Bioresearch Monitoring Program Managers for each Center within the Food and Drug Administration.

[2] Under 21 CFR 312.62(b), reference is made to records that are part of case histories as "supporting data"; the ICH *E6 Good Clinical Practice* consolidated guidance uses the term "source documents." For the purpose of this guidance, these terms describe the same information and have been used interchangeably.

[3] Human drugs include biological drugs.

[4] In August 2003, FDA issued the guidance for industry entitled *Part 11, Electronic Records; Electronic Signatures-Scope and Application* clarifying that the Agency intends to interpret the scope of part 11 narrowly and to exercise enforcement discretion with regard to part 11 requirements for validation, audit trails, record retention, and record copying. In 1996, the International Conference on Harmonisation of Technical Requirements for Registration of Pharmaceuticals for Human Use (ICH) issued *E6 Good Clinical Practice: Consolidated Guidance*.

FDA's guidance documents, including this guidance, do not establish legally enforceable responsibilities. Instead, guidances describe the Agency's current thinking on a topic and should be viewed only as recommendations, unless specific regulatory or statutory requirements are cited. The use of the word *should* in Agency guidances means that something is suggested or recommended, but not required.

II. BACKGROUND

There is an increasing use of computerized systems in clinical trials to generate and maintain source data and source documentation on each clinical trial subject. Such electronic source data and source documentation must meet the same fundamental elements of data quality (e.g., attributable, legible, contemporaneous, original,[5] and accurate) that are expected of paper records and must comply with all applicable statutory and regulatory requirements. FDA's acceptance of data from clinical trials for decision-making purposes depends on FDA's ability to verify the quality and integrity of the data during FDA on-site inspections and audits. (21 CFR 312, 511.1(b), and 812).

In March 1997, FDA issued 21 CFR part 11, which provides criteria for acceptance by FDA, under certain circumstances, of electronic records, electronic signatures, and handwritten signatures executed to electronic records as equivalent to paper records and handwritten signatures executed on paper. After the effective date of 21 CFR part 11, significant concerns regarding the interpretation and implementation of part 11 were raised by both FDA and industry. As a result, we decided to reexamine 21 CFR part 11 with the possibility of proposing additional rulemaking, and exercising enforcement discretion regarding enforcement of certain part 11 requirements in the interim.

This guidance finalizes the draft guidance for industry entitled *Computerized Systems Used in Clinical Trials,* dated September 2004 and supplements the guidance for industry entitled *Part 11, Electronic Records; Electronic Signatures – Scope and Application* (Scope and Application Guidance), dated August 2003. The Scope and Application Guidance clarified that the Agency intends to interpret the scope of part 11 narrowly and to exercise enforcement discretion with regard to part 11 requirements for validation, audit trails, record retention, and record copying. However, other Part 11 provisions remain in effect.

The approach outlined in the Scope and Application Guidance, which applies to electronic records generated as part of a clinical trial, should be followed until such time as Part 11 is amended.

[5] FDA is allowing original documents to be replaced by copies provided the copies are identical and have been verified as such (See, e.g., FDA Compliance Policy Guide # 7150.13). See Definitions section for a definition of original data.

III. SCOPE

The principles outlined in this guidance should be used for computerized systems that contain any data that are relied on by an applicant in support of a marketing application, including computerized laboratory information management systems that capture analytical results of tests conducted during a clinical trial. For example, the recommendations in this guidance would apply to computerized systems that create source documents (electronic records) that satisfy the requirements in 21 CFR 312.62(b) and 812.140(b), such as case histories. This guidance also applies to recorded source data transmitted from automated instruments directly to a computerized system (e.g., data from a chemistry autoanalyser or a Holter monitor to a laboratory information system). This guidance also applies when source documentation is created in hardcopy and later entered into a computerized system, recorded by direct entry into a computerized system, or automatically recorded by a computerized system (e.g., an ECG reading). The guidance does not apply to computerized medical devices that generate such data and that are otherwise regulated by FDA.

IV. RECOMMENDATIONS

This guidance provides the following recommendations regarding the use of computerized systems in clinical investigations.

A. Study Protocols

Each specific study protocol should identify each step at which a computerized system will be used to create, modify, maintain, archive, retrieve, or transmit source data. This information can be included in the protocol at the time the investigational new drug application (IND), Investigational Device Exemption (IDE), or Notice of Claimed Investigational Exemption for a New Animal Drug containing the protocols is submitted or at any time after the initial submission.

The computerized systems should be designed: (1) to satisfy the processes assigned to these systems for use in the specific study protocol (e.g., record data in metric units, blind the study), and (2) to prevent errors in data creation, modification, maintenance, archiving, retrieval, or transmission (e.g., inadvertently unblinding a study).

B. Standard Operating Procedures

There should be specific procedures and controls in place when using computerized systems to create, modify, maintain, or transmit electronic records, including when collecting source data at clinical trial sites. A list of recommended standard operating procedures (SOPs) is provided in Appendix A. Such SOPs should be maintained either on-site or be remotely accessible through electronic files as part of the specific study records, and the SOPs should be made available for use by personnel and for inspection by FDA.

C. Source Documentation and Retention

When original observations are entered directly into a computerized system, the electronic record is the source document. Under 21 CFR 312.62, 511.1(b)(7)(ii) and 812.140, the clinical investigator must retain records required to be maintained under part 312, § 511.1(b), and part 812, for a period of time specified in these regulations. This requirement applies to the retention of the original source document, or a copy of the source document.

When source data are transmitted from one system to another (e.g., from a personal data assistant to a sponsor's server), or entered directly into a remote computerized system (e.g., data are entered into a remote server via a computer terminal that is located at the clinical site), or an electrocardiogram at the clinical site is transmitted to the sponsor's computerized system, a copy of the data should be maintained at another location, typically at the clinical site but possibly at some other designated site. Copies should be made contemporaneously with data entry and should be preserved in an appropriate format, such as XML, PDF or paper formats.

D. Internal Security Safeguards

1. Limited Access

Access must be limited to authorized individuals (21 CFR 11.10(d). This requirement can be accomplished by the following recommendations. We recommend that each user of the system have an individual account. The user should log into that account at the beginning of a data entry session, input information (including changes) on the electronic record, and log out at the completion of data entry session. The system should be designed to limit the number of log-in attempts and to record unauthorized access log-in attempts.

Individuals should work only under their own password or other access key and not share these with others. The system should not allow an individual to log onto the system to provide another person access to the system. We also recommend that passwords or other access keys be changed at established intervals commensurate with a documented risk assessment.

When someone leaves a workstation, the person should log off the system. Alternatively, an automatic log off may be appropriate for long idle periods. For short periods of inactivity, we recommend that a type of automatic protection be installed against unauthorized data entry (e.g., an automatic screen saver can prevent data entry until a password is entered).

2. Audit Trails

It is important to keep track of all changes made to information in the electronic records that document activities related to the conduct of the trial (audit trails). The use of audit trails or other security measures helps to ensure that only authorized additions, deletions, or alterations of information in the electronic record have occurred and allows a means to reconstruct significant details about study conduct and source data collection necessary to verify the quality and integrity of data. Computer-generated, time-stamped audit trails or other security measures can also capture information related to the creation, modification, or deletion of electronic records and may be useful to ensure compliance with the appropriate regulation.

The need for audit trails should be determined based on a justified and documented risk assessment that takes into consideration circumstances surrounding system use, the likelihood that information might be compromised, and any system vulnerabilities. Should it be decided that audit trails or other appropriate security measures are needed to ensure electronic record integrity, personnel who create, modify, or delete electronic records should not be able to modify the documents or security measures used to track electronic record changes. Computer-generated, time-stamped electronic audits trails are the preferred method for tracking changes to electronic source documentation.

Audit trails or other security methods used to capture electronic record activities should describe when, by whom, and the reason changes were made to the electronic record. Original information should not be obscured though the use of audit trails or other security measures used to capture electronic record activities.

3. Date/Time Stamps

Controls should be established to ensure that the system's date and time are correct. The ability to change the date or time should be limited to authorized personnel, and such personnel should be notified if a system date or time discrepancy is detected. Any changes to date or time should always be documented. We do not expect documentation of time changes that systems make automatically to adjust to daylight savings time conventions.

We recommend that dates and times include the year, month, day, hour, and minute and encourage synchronization of systems to the date and time provided by international standard-setting agencies (e.g., U.S. National Institute of Standards and Technology provides information about universal time, coordinated (UTC)).

Computerized systems are likely to be used in multi-center clinical trials and may be located in different time zones. For systems that span different time zones, it is better to implement time stamps with a clear understanding of the time zone reference used. We recommend that system documentation explain time zone references as well as zone acronyms or other naming conventions.

E. External Security Safeguards

In addition to internal safeguards built into a computerized system, external safeguards should be put in place to ensure that access to the computerized system and to the data is restricted to authorized personnel. Staff should be kept thoroughly aware of system security measures and the importance of limiting access to authorized personnel.

Procedures and controls should be put in place to prevent the altering, browsing, querying, or reporting of data via external software applications that do not enter through the protective system software.

You should maintain a cumulative record that indicates, for any point in time, the names of authorized personnel, their titles, and a description of their access privileges. That record should be kept in the study documentation, accessible for use by appropriate study personnel and for inspection by FDA investigators.

We also recommend that controls be implemented to prevent, detect, and mitigate effects of computer viruses, worms, or other potentially harmful software code on study data and software.

F. Other System Features

1. Direct Entry of Data

We recommend that you incorporate prompts, flags, or other help features into your computerized system to encourage consistent use of clinical terminology and to alert the user to data that are out of acceptable range. You should not use programming features that automatically enter data into a field when the field is bypassed (default entries). However, you can use programming features that permit repopulation of information specific to the subject. To avoid falsification of data, you should perform a careful analysis in deciding whether and when to use software programming instructions that permit data fields to be automatically populated.

2. Retrieving Data

The computerized system should be designed in such a way that retrieved data regarding each individual subject in a study is attributable to that subject. Reconstruction of the source documentation is essential to FDA's review of the clinical study submitted to the Agency. Therefore, the information provided to FDA should fully describe and explain how source data were obtained and managed, and how electronic records were used to capture data.

It is not necessary to reprocess data from a study that can be fully reconstructed from available documentation. Therefore, the actual application software, operating systems, and software development tools involved in the processing of data or records need not be retained.

3. Dependability System Documentation

For each study, documentation should identify what software and hardware will be used to create, modify, maintain, archive, retrieve, or transmit clinical data. Although it need not be submitted to FDA, this documentation should be retained as part of the study records and be available for inspection by FDA (either on-site or remotely accessible).

4. System Controls

When electronic formats are the only ones used to create and preserve electronic records, sufficient backup and recovery procedures should be designed to protect against data loss. Records should regularly be backed up in a procedure that would prevent a catastrophic loss and ensure the quality and integrity of the data. Records should be stored at a secure location specified in the SOP. Storage should typically be offsite or in a building separate from the original records.

We recommend that you maintain backup and recovery logs to facilitate an assessment of the nature and scope of data loss resulting from a system failure.

5. Change Controls

The integrity of the data and the integrity of the protocols should be maintained when making changes to the computerized system, such as software upgrades, including security and performance patches, equipment, or component replacement, or new instrumentation. The effects of any changes to the system should be evaluated and some should be validated depending on risk. Changes that exceed previously established operational limits or design specifications should be validated. Finally, all changes to the system should be documented.

G. Training of Personnel

Those who use computerized systems must determine that individuals (e.g., employees, contractors) who develop, maintain, or use computerized systems have the education, training and experience necessary to perform their assigned tasks (21 CFR 11.10(i)).

Training should be provided to individuals in the specific operations with regard to computerized systems that they are to perform. Training should be conducted by qualified individuals on a continuing basis, as needed, to ensure familiarity with the computerized system and with any changes to the system during the course of the study.

We recommend that computer education, training, and experience be documented.

DEFINITIONS

The following is a list of definitions for terms used in, and for the purposes of, this guidance document.

Audit Trail: For the purpose of this guidance, an *audit trail* is a process that captures details such as additions, deletions, or alterations of information in an electronic record without obliterating the original record. An audit trail facilitates the reconstruction of the course of such details relating to the electronic record.

Certified Copy: A *certified copy* is a copy of original information that has been verified, as indicated by a dated signature, as an exact copy having all of the same attributes and information as the original.

Computerized System: A *computerized system* includes computer hardware, software, and associated documents (e.g., user manual) that create, modify, maintain, archive, retrieve, or transmit in digital form information related to the conduct of a clinical trial.

Direct Entry: *Direct entry* is recording data where an electronic record is the original means of capturing the data. Examples are the keying by an individual of original observations into a system, or automatic recording by the system of the output of a balance that measures subject's body weight.

Electronic Record: An *electronic record* is any combination of text, graphics, data, audio, pictorial, or other information representation in digital form that is created, modified, maintained, archived, retrieved, or distributed by a computer system.

Original data: For the purpose of this guidance, *original data* are those values that represent the first recording of study data. FDA is allowing original documents and the original data recorded on those documents to be replaced by copies provided the copies are identical and have been verified as such (see FDA Compliance Policy Guide # 7150.13).

Source Documents: Original documents and records including, but not limited to, hospital records, clinical and office charts, laboratory notes, memoranda, subjects' diaries or evaluation checklists, pharmacy dispensing records, recorded data from automated instruments, copies or transcriptions certified after verification as being accurate and complete, microfiches, photographic negatives, microfilm or magnetic media, x-rays, subject files, and records kept at the pharmacy, at the laboratories, and at medico-technical departments involved in a clinical trial.

Transmit: *Transmit* is to transfer data within or among clinical study sites, contract research organizations, data management centers, sponsors, or to FDA.

REFERENCES

FDA, *21 CFR Part 11, "*Electronic Records; Electronic Signatures; Final Rule.*" Federal Register* Vol. 62, No. 54, 13429, March 20, 1997.

FDA, *Compliance Program Guidance Manual*, "Compliance Program 7348.810 – Bioresearch Monitoring - Sponsors, Contract Research Organizations and Monitors," February 21, 2001.

FDA, *Compliance Program Guidance Manual*, "Compliance Program 7348.811 - Bioresearch Monitoring - Clinical Investigators," September 30, 2000.

FDA, *Good Clinical Practice VICH GL9.*

FDA, *Guideline for the Monitoring of Clinical Investigations*.

FDA, *Information Sheets for Institutional Review Boards and Clinical Investigators*.

http://www.fda.gov/ic/ohrt/irbs/default.htm

FDA, *E6 Good Clinical Practice: Consolidated Guidance*. http://www.fda.gov/cder/guidance/959fnl.pdf.

FDA, *Part 11, Electronic Records; Electronic Signatures — Scope and Application*, 2003.

FDA, *General Principles of Software Validation; Guidance for Industry and FDA Staff.*

APPENDIX A

STANDARD OPERATING PROCEDURES

Standard operating procedures (SOPs) and documentation pertinent to the use of a computerized system should be made available for use by appropriate study personnel at the clinical site or remotely and for inspection by FDA. 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

SOFTWARECPR®
CRISIS PREVENTION AND RECOVERY, LLC

Software Validation Compliance Assessment Job Aide: Clinical Trials Software Guidance Checklist

Mary Decareau, Oct. 2, 2007

Jointly issued by FDA's CDER, CBER, CDRH, ORA, CVM, and CFSAN "Guidance for Industry: Computerized Systems Used in Clinical Investigations" issued as final in May 2007. Supersedes prior guidance of the same name.

This document describes FDA's expectations for validation of systems used to support Clinical Trials including data entry, system features, security, system dependability, system controls, personnel training, and provision for inspection by FDA . Although written for computer systems to be used in clinical trials there is a great deal of information in this document that is relevant to validation of many types of computer systems that are subject to GMP requirements for validation. The checklist below is a job aide for those involved in assessing conformance to this guidance or attempting to implement and validate a system for clinical trials use. It is not to be used in place of a thorough understanding of the guidance document and FDA enforcement practices.

Purpose

- This assessment aide serves as an **informal** checklist for reference.
- It is not to be used as a mindless checklist.
- Not all questions are relevant to all interviewees and projects.
- Questions are to be spread over a number of interviewees and not all asked of each individual.
- An assessment would not normally proceed in the order of the sections of this job aide
- and many issues would be addressed at one time with the appropriate individual(s)
- Notes can be kept in the spaces provided or separately as preferred.
- THIS IS ONLY A GUIDE and REMINDER and does not reflect on the form or content of any final report or recommendations.
- Other checklists (e.g., validation, Part 11, QS reg, GMP) should be used in conjuction with this checklist as applicable.
- THIS DOES NOT SERVE as a list of best practices or a standard assessment scheme.

At the end of the checklist are two items from the guidance itself and a third item that is not and provides some general tips:

- The Table of Contents from the guidance
- Section III, Scope, from the guidance as this provides the Agency’s basic intent which is useful in helping to interpret and set priorities for the specific requirements

IV. A Study Protocols

Checklist item	**Present? (Y/N)**	**Comments**
Study Protocols exist that define each step at which a computerized system will be used to create, modify, maintain, archive, retrieve or transmit source data.		
Software used should meet its purpose and include features that prevent errors in data creation, modification, maintenance, archiving, retrieval or transmission.		

B. Standard Operating Procedures

Checklist item	**Present? (Y/N)**	**Comments**
SOPs should be maintained on site or be remotely accessible and be available for personnel and for inspection by FDA.		
Appendix A Recommended SOPs are listed below:		
System Setup/Installation (including the description and specific use of software, harware related to the physical environment)		
System operating manual		
Validation and functionality testing		
Data Collection and Handling (including data archiving, audit trails and risk assessment)		
System Maintenance (including system decomissioning)		
System security measures		
Change Control		
Data Backup, Recovery and Contingency Planning		
Alternative recording methods (if system is unavailable)		
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		

C. Source Documentation and Retention

Checklist item	**Present? (Y/N)**	**Comments**
Records are retained as specified in 21 CFR 312.62, 511.1(b)(7)(ii) and part 812. This applies to the original source document or a copy of the source document.		
When data is transmitted, a copy must be maintained at another location, typically at the clinical site from which the data was transmitted.		

Checklist item	Present? (Y/N)	Comments
Copies should be made contemporaneously with data entry and should be preserved in an appropriate format such as XML, PDF or paper.		
Retain (or have access to) old systems or transcribe data*		
Scripts/query logic documented and validated*		
Certification Statement Sent to FDA for e-Signatures*		

* Not in guidance but consider to meet the intent

D. Internal Security Safeguards

1) Limited Access

Checklist item	Present? (Y/N)	Comments
Access to systems is limited to authorized users. For example, with logins for each user.		
Number of log in attempts are limited.		
Unauthorized login attempts are recorded.		
Passwords are required to be changed periodically (according to risk assessment)		
When the system is left idle users should logoff – is this in a procedures?, or the user is automatically logged off or a screen saver prevents data entry until a password is entered.		

2) Audit Trails

Checklist item	Present? (Y/N)	Comments
Audit trails or alternative security measures are in use, or the need for audit trails is determined based on a justified and documented risk assessment that takes into consideration circumstances surrounding system use, the likelihood that information might be compromised, and any system vulnerabilities.		
Time-stamped audit trails exist – the creation, modification, deletion of data can be traced to the person making the change and the date and time the change was made.		
The reason for modification is captured.		
Audit trails are readable & copiable by FDA for retention period at all locations.		
The original information must not be obscured.		
Audit trails must be secure – users are not able to modify the audit trail.		

3) Date/Time Stamps

Checklist item	**Present? (Y/N)**	**Comments**
Controls exist to ensure date/time are correct.		
Date/time changes can only be made by authorized users.		
Changes to date/time are documented (does not include automatic daylight savings conversion)		
Dates and times include year, month, day, hour and minute.		
Time is synched to trusted 3rd parties		
If systems span time zones, the local time can be derived from central server		
Time zone conventions and abbreviations are documented.		

E. External Security

Checklist item	**Present? (Y/N)**	**Comments**
External physical security exists to ensure that access to the computerized system and the data is restricted to authorized personnel.		
Staff is trained in security measures, and training documentation exists.		
No access to data is possible through interfaces or other software that bypasses security. Controls to prevent external applications from bypassing security for altering, browsing, querying or reporting of data.		
Cumulative access list records by date, including names **and titles**, and access rights exists for each study.		
Software and data on shared use systems should be protected and controlled*		
Virus and other threat protection is employed.		

* Not in guidance but consider to meet the intent

F. Other System Features

1. Direct Entry of Data

Checklist item	Present? (Y/N)	Comments
Use prompts, flags, range checking, etc. exist to ensure that data is properly entered.		
System does not allow default data entry if a field is bypassed		
If the system uses features to repopulate data specific to the subject, care must be taken to ensure that data is not falsified.		

2. Retrieving Data

Checklist item	Present? (Y/N)	Comments
Retrieved data must be attributable to each subject.		
Reconstruction of source documentation - Must be able to explain how source data was obtained and managed and how electronic records were used to capture data.		
If data can be fully reconstructed from available documentation, it is not necessary to keep the application software, operating systems, software development tools, etc. Otherwise, must have the means to reconstruct the data.		

3. Dependability System Documentation

Checklist item	Present? (Y/N)	Comments
Systems documentation available at trial site or remotely accessible (including a description of the HW&SW used to create, modify, maintain, archive, retreive or transmit data) and available for inspection.		

4. System Controls

Checklist item	Present? (Y/N)	Comments
Backup and recovery procedures exist.		
Offsite secure storage of Backup as specified in the SOP.		
Backup and recovery Logs are maintained.		

5. Change Controls

Checklist item	Present? (Y/N)	Comments
Changes such as software upgrades, security and performance patches, equipment or component replacement or		

Checklist item	Present? (Y/N)	Comments
new instrumentation are documented.		
Changes that exceed previously established operational limits or design specifications are validated.		

G. Training

Checklist item	Present? (Y/N)	Comments
Documentation of computer education, training and experience exists for: Users, developers, maintenance and administration staff		
Training on a continuing/periodic basis to ensure familiarity and update for changes		

This checklist is for items in the guidance related to the computer system. It does not include items related to trials site responsibilities or other non-computer related information.

TABLE OF CONTENTS

III. SCOPE

The principles outlined in this guidance should be used for computerized systems that contain any data that are relied on by an applicant in support of a marketing application, including computerized laboratory information management systems that capture analytical results of tests conducted during a clinical trial. For example, the recommendations in this guidance would apply to computerized systems that create source documents (electronic records) that satisfy the requirements in 21 CFR 312.62(b) and 812.140(b), such as case histories. This guidance also applies to recorded source data transmitted from automated instruments directly to a computerized system (e.g., data from a chemistry autoanalyser or a Holter monitor to a laboratory information system). This guidance also applies when source documentation is created in hardcopy and later entered into a computerized system, recorded by direct entry into a computerized system, or automatically recorded by a computerized system (e.g., an ECG reading). The guidance does not apply to computerized medical devices that generate such data and that are otherwise regulated by FDA.

Validation Tips for Computerized Systems used in Clinical Investigations

The Clinical Investigation site should have performed an assessment of the software provider and have validated the software for their use based on the assessment results. If the software provider did not have good validation records, then the software probably needs more in-depth validation.

Data Entry and modification–

- Critical data may be required to be entered twice, or twice by two different operators. Make sure that the validation testing covers that the data can be entered twice for both successful and failure conditions.

- Verify that units of measure are clear in the software, and any conversions have been verified.

- Verify that validation has checked for two people modifying the same record at the same time. Related design item - the software should be designed so that data is committed as one entity if more than one table or record are involved.

- Validation tests should check record commitments to ensure that not only is the data saved, but the time stamp and any other important information are saved for all records, including audit trails.

Data Retrieval and Query –

- Data retrieval and query functions may need special attention if they perform critical tasks (for example, if important clinical decisions may be made based on the query results). Validation testing may need to cover checking that queries work both from the application and by verifying with separate queries performed on the database.

Data Integrity –

- The system design should ensure that data that is related to a sample (for example, a blood or tissue sample) or to a Subject, must be kept intact. For example, if a sample has results associated with it, the design should ensure that those results cannot inadvertently be associated with the incorrect sample.

- Audit trails need to be thoroughly tested, especially that the time/date stamps, and user making the change are correctly recorded. Viewing or reporting the audit trail need to be covered.

- Validation of importing/exporting data from/to other systems or from instruments should cover “good” data and incorrectly formatted or bad data. This type of validation should also cover importing/exporting the maximum amount of data.

Data Anonymization –

- Systems may include features to anonymize or “blind” data. There may be two steps, one to assign new IDs to data but still maintaining a link to the Subject, and the second that completely removes the link to the subject (this is not usually reversible) The validation of these features should be carefully scrutinized.

Security –

- It is important to ensure that the data cannot be accessed by other means, for example data stored in an Oracle database is not accessible to unauthorized users via SQL+ or other tools. An authorized user should be a Study director, Database Administrator or similar user. Regular users should not be able to access data through these tools.

Changes –

- There should be procedures in place governing the changes to the system/software AND to the data. Database administrators may make changes to "fix" problems, be sure that these are documented and covered under procedures.
- Changes involving a migration of data to a new database should have extensive validation – verification for every column in each table, for example. This validation should also include verifying the migration of various types of data with actual data or a realistic recreation of the data.

Backup/Recovery –

- Procedures should include "disaster recovery", recovering from a complete loss of the computer system (how to start from scratch, including which hardware to purchase, what needs to be loaded on it, etc.) The disaster recovery plan needs to be in a format that can be accessed if the disaster happens.
- Backup/recovery procedures should have been tested.

Guidance for Industry

Part 11, Electronic Records; Electronic Signatures — Scope and Application

U.S. Department of Health and Human Services
Food and Drug Administration
Center for Drug Evaluation and Research (CDER)
Center for Biologics Evaluation and Research (CBER)
Center for Devices and Radiological Health (CDRH)
Center for Food Safety and Applied Nutrition (CFSAN)
Center for Veterinary Medicine (CVM)
Office of Regulatory Affairs (ORA)

August 2003
Pharmaceutical CGMPs

Guidance for Industry

Part 11, Electronic Records; Electronic Signatures — Scope and Application

Division of Drug Information, HFD-240
Center for Drug Evaluation and Research (CDER)
(Tel) 301-827-4573
http://www.fda.gov/cder/guidance/index.htm
or
Office of Communication, Training and
Manufacturers Assistance, HFM-40
Center for Biologics Evaluation and Research (CBER)
http://www.fda.gov/cber/guidelines.htm
Phone: the Voice Information System at 800-835-4709 or 301-827-1800
or
Communications Staff (HFV-12),
Center for Veterinary Medicine (CVM)
(Tel) 301-594-1755
http://www.fda.gov/cvm/guidance/guidance.html
or
Division of Small Manufacturers Assistance (HFZ-220)
http://www.fda.gov/cdrh/ggpmain.html
Manufacturers Assistance Phone Number: 800.638.2041 or 301.443.6597
Internt'l Staff Phone: 301.827.3993
or
Center for Food Safety and Applied Nutrition (CFSAN)
http://www.cfsan.fda.gov/~dms/guidance.html.

U.S. Department of Health and Human Services
Food and Drug Administration
Center for Drug Evaluation and Research (CDER)
Center for Biologics Evaluation and Research (CBER)
Center for Devices and Radiological Health (CDRH)
Center for Food Safety and Applied Nutrition (CFSAN)
Center for Veterinary Medicine (CVM)
Office of Regulatory Affairs (ORA)

August 2003
Pharmaceutical CGMPs

TABLE OF CONTENTS

Guidance for Industry[1]
Part 11, Electronic Records; Electronic Signatures — Scope and Application

This guidance represents the Food and Drug Administration's (FDA's) current thinking on this topic. It does not create or confer any rights for or on any person and does not operate to bind FDA or the public. You can use an alternative approach if the approach satisfies the requirements of the applicable statutes and regulations. If you want to discuss an alternative approach, contact the FDA staff responsible for implementing this guidance. If you cannot identify the appropriate FDA staff, call the appropriate number listed on the title page of this guidance.

I. INTRODUCTION

This guidance is intended to describe the Food and Drug Administration's (FDA's) current thinking regarding the scope and application of part 11 of Title 21 of the Code of Federal Regulations; Electronic Records; Electronic Signatures (21 CFR Part 11).[2]

This document provides guidance to persons who, in fulfillment of a requirement in a statute or another part of FDA's regulations to maintain records or submit information to FDA,[3] have chosen to maintain the records or submit designated information electronically and, as a result, have become subject to part 11. Part 11 applies to records in electronic form that are created, modified, maintained, archived, retrieved, or transmitted under any records requirements set forth in Agency regulations. Part 11 also applies to electronic records submitted to the Agency under the Federal Food, Drug, and Cosmetic Act (the Act) and the Public Health Service Act (the PHS Act), even if such records are not specifically identified in Agency regulations (§ 11.1). The underlying requirements set forth in the Act, PHS Act, and FDA regulations (other than part 11) are referred to in this guidance document as *predicate rules.*

[1] This guidance has been prepared by the Office of Compliance in the Center for Drug Evaluation and Research (CDER) in consultation with the other Agency centers and the Office of Regulatory Affairs at the Food and Drug Administration.

[2] 62 FR 13430

[3] These requirements include, for example, certain provisions of the Current Good Manufacturing Practice regulations (21 CFR Part 211), the Quality System regulation (21 CFR Part 820), and the Good Laboratory Practice for Nonclinical Laboratory Studies regulations (21 CFR Part 58).

As an outgrowth of its current good manufacturing practice (CGMP) initiative for human and animal drugs and biologics,[4] FDA is re-examining part 11 as it applies to all FDA regulated products. We anticipate initiating rulemaking to change part 11 as a result of that re-examination. This guidance explains that we will narrowly interpret the scope of part 11. While the re-examination of part 11 is under way, we intend to exercise enforcement discretion with respect to certain part 11 requirements. That is, we do not intend to take enforcement action to enforce compliance with the validation, audit trail, record retention, and record copying requirements of part 11 as explained in this guidance. However, records must still be maintained or submitted in accordance with the underlying predicate rules, and the Agency can take regulatory action for noncompliance with such predicate rules.

In addition, we intend to exercise enforcement discretion and do not intend to take (or recommend) action to enforce any part 11 requirements with regard to systems that were operational before August 20, 1997, the effective date of part 11 (commonly known as legacy systems) under the circumstances described in section III.C.3 of this guidance.

Note that part 11 remains in effect and that this exercise of enforcement discretion applies only as identified in this guidance.

FDA's guidance documents, including this guidance, do not establish legally enforceable responsibilities. Instead, guidances describe the Agency's current thinking on a topic and should be viewed only as recommendations, unless specific regulatory or statutory requirements are cited. The use of the word *should* in Agency guidances means that something is suggested or recommended, but not required.

II. BACKGROUND

In March of 1997, FDA issued final part 11 regulations that provide criteria for acceptance by FDA, under certain circumstances, of electronic records, electronic signatures, and handwritten signatures executed to electronic records as equivalent to paper records and handwritten signatures executed on paper. These regulations, which apply to all FDA program areas, were intended to permit the widest possible use of electronic technology, compatible with FDA's responsibility to protect the public health.

After part 11 became effective in August 1997, significant discussions ensued among industry, contractors, and the Agency concerning the interpretation and implementation of the regulations. FDA has (1) spoken about part 11 at many conferences and met numerous times with an industry coalition and other interested parties in an effort to hear more about potential part 11 issues; (2) published a compliance policy guide, CPG 7153.17: Enforcement Policy: 21 CFR Part 11; Electronic Records; Electronic Signatures; and (3) published numerous draft guidance documents including the following:

[4] See *Pharmaceutical CGMPs for the 21st Century: A Risk-Based Approach; A Science and Risk-Based Approach to Product Quality Regulation Incorporating an Integrated Quality Systems Approach* at www.fda.gov/oc/guidance/gmp.html.

- *21 CFR Part 11; Electronic Records; Electronic Signatures, Validation*
- *21 CFR Part 11; Electronic Records; Electronic Signatures, Glossary of Terms*
- *21 CFR Part 11; Electronic Records; Electronic Signatures, Time Stamps*
- *21 CFR Part 11; Electronic Records; Electronic Signatures, Maintenance of Electronic Records*
- *21 CFR Part 11; Electronic Records; Electronic Signatures, Electronic Copies of Electronic Records*

Throughout all of these communications, concerns have been raised that some interpretations of the part 11 requirements would (1) unnecessarily restrict the use of electronic technology in a manner that is inconsistent with FDA's stated intent in issuing the rule, (2) significantly increase the costs of compliance to an extent that was not contemplated at the time the rule was drafted, and (3) discourage innovation and technological advances without providing a significant public health benefit. These concerns have been raised particularly in the areas of part 11 requirements for validation, audit trails, record retention, record copying, and legacy systems.

As a result of these concerns, we decided to review the part 11 documents and related issues, particularly in light of the Agency's CGMP initiative. In the *Federal Register* of February 4, 2003 (68 FR 5645), we announced the withdrawal of the draft guidance for industry, *21 CFR Part 11; Electronic Records; Electronic Signatures, Electronic Copies of Electronic Records*. We had decided we wanted to minimize industry time spent reviewing and commenting on the draft guidance when that draft guidance may no longer represent our approach under the CGMP initiative. Then, in the *Federal Register* of February 25, 2003 (68 FR 8775), we announced the withdrawal of the part 11 draft guidance documents on validation, glossary of terms, time stamps,[5] maintenance of electronic records, and CPG 7153.17. We received valuable public comments on these draft guidances, and we plan to use that information to help with future decision-making with respect to part 11. We do not intend to re-issue these draft guidance documents or the CPG.

We are now re-examining part 11, and we anticipate initiating rulemaking to revise provisions of that regulation. To avoid unnecessary resource expenditures to comply with part 11 requirements, we are issuing this guidance to describe how we intend to exercise enforcement discretion with regard to certain part 11 requirements during the re-examination of part 11. As mentioned previously, part 11 remains in effect during this re-examination period.

III. DISCUSSION

A. Overall Approach to Part 11 Requirements

[5] Although we withdrew the draft guidance on time stamps, our current thinking has not changed in that when using time stamps for systems that span different time zones, we do not expect you to record the signer's local time. When using time stamps, they should be implemented with a clear understanding of the time zone reference used. In such instances, system documentation should explain time zone references as well as zone acronyms or other naming conventions.

As described in more detail below, the approach outlined in this guidance is based on three main elements:

- Part 11 will be interpreted narrowly; we are now clarifying that fewer records will be considered subject to part 11.
- For those records that remain subject to part 11, we intend to exercise enforcement discretion with regard to part 11 requirements for validation, audit trails, record retention, and record copying in the manner described in this guidance and with regard to all part 11 requirements for systems that were operational before the effective date of part 11 (also known as legacy systems).
- We will enforce all predicate rule requirements, including predicate rule record and recordkeeping requirements.

It is important to note that FDA's exercise of enforcement discretion as described in this guidance is limited to specified part 11 requirements (setting aside legacy systems, as to which the extent of enforcement discretion, under certain circumstances, will be more broad). We intend to enforce all other provisions of part 11 including, but not limited to, certain controls for closed systems in § 11.10. For example, we intend to enforce provisions related to the following controls and requirements:

- limiting system access to authorized individuals
- use of operational system checks
- use of authority checks
- use of device checks
- determination that persons who develop, maintain, or use electronic systems have the education, training, and experience to perform their assigned tasks
- establishment of and adherence to written policies that hold individuals accountable for actions initiated under their electronic signatures
- appropriate controls over systems documentation
- controls for open systems corresponding to controls for closed systems bulleted above (§ 11.30)
- requirements related to electronic signatures (e.g., §§ 11.50, 11.70, 11.100, 11.200, and 11.300)

We expect continued compliance with these provisions, and we will continue to enforce them. Furthermore, persons must comply with applicable predicate rules, and records that are required to be maintained or submitted must remain secure and reliable in accordance with the predicate rules.

B. Details of Approach – Scope of Part 11

1. Narrow Interpretation of Scope

We understand that there is some confusion about the scope of part 11. Some have understood the scope of part 11 to be very broad. We believe that some of those broad interpretations could lead to unnecessary controls and costs and could discourage innovation and technological advances without providing added benefit to the public health. As a result, we want to clarify that the Agency intends to interpret the scope of part 11 narrowly.

Under the narrow interpretation of the scope of part 11, with respect to records required to be maintained under predicate rules or submitted to FDA, when persons choose to use records in electronic format in place of paper format, part 11 would apply. On the other hand, when persons use computers to generate paper printouts of electronic records, and those paper records meet all the requirements of the applicable predicate rules and persons rely on the paper records to perform their regulated activities, FDA would generally not consider persons to be "using electronic records in lieu of paper records" under §§ 11.2(a) and 11.2(b). In these instances, the use of computer systems in the generation of paper records would not trigger part 11.

2. Definition of Part 11 Records

Under this narrow interpretation, FDA considers part 11 to be applicable to the following records or signatures in electronic format (part 11 records or signatures):

- Records that are required to be maintained under predicate rule requirements and that are maintained in electronic format *in place of paper format*. On the other hand, records (and any associated signatures) that are not required to be retained under predicate rules, but that are nonetheless maintained in electronic format, are not part 11 records.

  We recommend that you determine, based on the predicate rules, whether specific records are part 11 records. We recommend that you document such decisions.

- Records that are required to be maintained under predicate rules, that are maintained in electronic format *in addition to paper format*, and that *are relied on to perform regulated activities*.

  In some cases, actual business practices may dictate whether you are *using* electronic records instead of paper records under § 11.2(a). For example, if a record is required to be maintained under a predicate rule and you use a computer to generate a paper printout of the electronic records, but you nonetheless rely on the electronic record to perform regulated activities, the Agency may consider you to be *using* the electronic record instead of the paper record. That is, the Agency may take your business practices into account in determining whether part 11 applies.

  Accordingly, we recommend that, for each record required to be maintained under predicate rules, you determine in advance whether you plan to rely on the electronic record or paper record to perform regulated activities. We recommend that you document this decision (e.g., in a Standard Operating Procedure (SOP), or specification document).

- Records submitted to FDA, under predicate rules (even if such records are not specifically identified in Agency regulations) in electronic format (assuming the records have been identified in docket number 92S-0251 as the types of submissions the Agency accepts in electronic format). However, a record that is not itself submitted, but is used in generating a submission, is not a part 11 record unless it is otherwise required to be maintained under a predicate rule and it is maintained in electronic format.

- Electronic signatures that are intended to be the equivalent of handwritten signatures, initials, and other general signings required by predicate rules. Part 11 signatures include electronic signatures that are used, for example, to document the fact that certain events or actions occurred in accordance with the predicate rule (e.g. *approved*, *reviewed*, and *verified*).

C. Approach to Specific Part 11 Requirements

1. Validation

The Agency intends to exercise enforcement discretion regarding specific part 11 requirements for validation of computerized systems (§ 11.10(a) and corresponding requirements in § 11.30). Although persons must still comply with all applicable predicate rule requirements for validation (e.g., 21 CFR 820.70(i)), this guidance should not be read to impose any additional requirements for validation.

We suggest that your decision to validate computerized systems, and the extent of the validation, take into account the impact the systems have on your ability to meet predicate rule requirements. You should also consider the impact those systems might have on the accuracy, reliability, integrity, availability, and authenticity of required records and signatures. Even if there is no predicate rule requirement to validate a system, in some instances it may still be important to validate the system.

We recommend that you base your approach on a justified and documented risk assessment and a determination of the potential of the system to affect product quality and safety, and record integrity. For instance, validation would not be important for a word processor used only to generate SOPs.

For further guidance on validation of computerized systems, see FDA's guidance for industry and FDA staff *General Principles of Software Validation* and also industry guidance such as the *GAMP 4 Guide* (See References).

2. Audit Trail

The Agency intends to exercise enforcement discretion regarding specific part 11 requirements related to computer-generated, time-stamped audit trails (§ 11.10 (e), (k)(2) and any corresponding requirement in §11.30). Persons must still comply with all applicable predicate rule requirements related to documentation of, for example, date (e.g., § 58.130(e)), time, or sequencing of events, as well as any requirements for ensuring that changes to records do not obscure previous entries.

Even if there are no predicate rule requirements to document, for example, date, time, or sequence of events in a particular instance, it may nonetheless be important to have audit trails or other physical, logical, or procedural security measures in place to ensure the trustworthiness and reliability of the records.[6] We recommend that you base your decision on whether to apply audit trails, or other appropriate measures, on the need to comply with predicate rule requirements, a justified and documented risk assessment, and a determination of the potential effect on product quality and safety and record integrity. We suggest that you apply appropriate controls based on such an assessment. Audit trails can be particularly appropriate when users are expected to create, modify, or delete regulated records during normal operation.

3. Legacy Systems[7]

The Agency intends to exercise enforcement discretion with respect to all part 11 requirements for systems that otherwise were operational prior to August 20, 1997, the effective date of part 11, under the circumstances specified below.

This means that the Agency does not intend to take enforcement action to enforce compliance with any part 11 requirements if all the following criteria are met for a specific system:

- The system was operational before the effective date.
- The system met all applicable predicate rule requirements before the effective date.
- The system currently meets all applicable predicate rule requirements.
- You have documented evidence and justification that the system is fit for its intended use (including having an acceptable level of record security and integrity, if applicable).

If a system has been changed since August 20, 1997, and if the changes would prevent the system from meeting predicate rule requirements, Part 11 controls should be applied to Part 11 records and signatures pursuant to the enforcement policy expressed in this guidance.

4. Copies of Records

The Agency intends to exercise enforcement discretion with regard to specific part 11 requirements for generating copies of records (§ 11.10 (b) and any corresponding requirement in §11.30). You should provide an investigator with reasonable and useful access to records during an inspection. All records held by you are subject to inspection in accordance with predicate rules (e.g., §§ 211.180(c), (d), and 108.35(c)(3)(ii)).

We recommend that you supply copies of electronic records by:

- Producing copies of records held in common portable formats when records are maintained in these formats
- Using established automated conversion or export methods, where available, to make copies in a more common format (examples of such formats include, but are not limited to, PDF, XML, or SGML)

[6] Various guidance documents on information security are available (see References).

[7] In this guidance document, we use the term *legacy system* to describe systems already in operation before the effective date of part 11.

In each case, we recommend that the copying process used produces copies that preserve the content and meaning of the record. If you have the ability to search, sort, or trend part 11 records, copies given to the Agency should provide the same capability if it is reasonable and technically feasible. You should allow inspection, review, and copying of records in a human readable form at your site using your hardware and following your established procedures and techniques for accessing records.

5. Record Retention

The Agency intends to exercise enforcement discretion with regard to the part 11 requirements for the protection of records to enable their accurate and ready retrieval throughout the records retention period (§ 11.10 (c) and any corresponding requirement in §11.30). Persons must still comply with all applicable predicate rule requirements for record retention and availability (e.g., §§ 211.180(c),(d), 108.25(g), and 108.35(h)).

We suggest that your decision on how to maintain records be based on predicate rule requirements and that you base your decision on a justified and documented risk assessment and a determination of the value of the records over time.

FDA does not intend to object if you decide to archive required records in electronic format to nonelectronic media such as microfilm, microfiche, and paper, or to a standard electronic file format (examples of such formats include, but are not limited to, PDF, XML, or SGML). Persons must still comply with all predicate rule requirements, and the records themselves and any copies of the required records should preserve their content and meaning. As long as predicate rule requirements are fully satisfied and the content and meaning of the records are preserved and archived, you can delete the electronic version of the records. In addition, paper and electronic record and signature components can co-exist (i.e., a hybrid[8] situation) as long as predicate rule requirements are met and the content and meaning of those records are preserved.

[8] Examples of hybrid situations include combinations of paper records (or other nonelectronic media) and electronic records, paper records and electronic signatures, or handwritten signatures executed to electronic records.

IV. REFERENCES

Food and Drug Administration References

1. *Glossary of Computerized System and Software Development Terminolo*gy (Division of Field Investigations, Office of Regional Operations, Office of Regulatory Affairs, FDA 1995) (http://www. fda.gov/ora/inspect_ref/igs/gloss.html)

2. *General Principles of Software Validation; Final Guidance for Industry and FDA Staff* (FDA, Center for Devices and Radiological Health, Center for Biologics Evaluation and Research, 2002) (http://www.fda.gov/cdrh/comp/guidance/938.html)

3. *Guidance for Industry, FDA Reviewers, and Compliance on Off-The-Shelf Software Use in Medical Devices* (FDA, Center for Devices and Radiological Health, 1999) (http://www.fda.gov/cdrh/ode/guidance/585.html)

4. *Pharmaceutical CGMPs for the 21st Century: A Risk-Based Approach; A Science and Risk-Based Approach to Product Quality Regulation Incorporating an Integrated Quality Systems Approach* (FDA 2002) (http://www.fda.gov/oc/guidance/gmp.html)

Industry References

1. *The Good Automated Manufacturing Practice (GAMP) Guide for Validation of Automated Systems*, *GAMP 4* (ISPE/GAMP Forum, 2001) (http://www.ispe.org/gamp/)

2. ISO/IEC 17799:2000 (BS 7799:2000) Information technology – Code of practice for information security management (ISO/IEC, 2000)

3. ISO 14971:2002 Medical Devices- Application of risk management to medical devices (ISO, 2001)

U.S. Food and Drug Administration

ORA OFFICE OF REGULATORY AFFAIRS

Office of Regulatory Affairs
Compliance References:
Bioresearch Monitoring (BIMO)

Compliance Program 7348.808 Contents:

ATTACHMENT A

Computerized Systems

The intent of this attachment is to collect, in one place, references to computer systems found throughout Part III. Computer systems and operations should be thoroughly covered during inspection of any facility. No additional reporting is required under this Attachment.

In August 1997, the Agency’s regulation on electronic signatures and electronic recordkeeping became effective. The Regulation, at 21 CFR Part 11, describes the technical and procedural requirements that must be met if a firm chooses to maintain records electronically and/or use electronic signatures. Part 11 works in conjunction with other FDA regulations and laws that require recordkeeping. Those regulations and laws ("predicate rules’) establish requirements for record content, signing, and retention.

Certain older electronic systems may not have been in full compliance with Part 11 by August 1997 and modification to these so called "legacy systems" may take more time. Part 11 does not grandfather legacy systems and FDA expects that firms using legacy systems are taking steps to achieve full compliance with Part 11.

If a firm is keeping electronic records or using electronic signatures, **determine** if they are in compliance with 21 CFR Part 11. **Determine** the depth of part 11 coverage on a case by case basis, in light of initial findings and program resources. At a minimum ensure that: (1) the firm has prepared a corrective action plan for achieving full compliance with part 11 requirements, and is making progress toward completing that plan in a timely manner; (2) accurate and complete electronic and human readable copies of electronic records, suitable for review, are made available; and (3) employees are held accountable and responsible for actions taken under their electronic signatures. If initial findings indicate the firm’s electronic records and/or electronic signatures may not be trustworthy and reliable, or when electronic recordkeeping systems inhibit meaningful FDA inspection, a more detailed evaluation may be warranted. Districts should consult with center compliance officers and the Office of Enforcement (HFC-240) in assessing the need for, and potential depth of, more detailed part 11 coverage. When substantial and significant part 11 deviations exist, FDA will not accept use of electronic records and electronic signatures to meet the requirements of the applicable predicate rule. See Compliance Policy Guide (CGP), Sec. 160.850.

See IOM sections 594.1 and 527.3 for procedures for collecting and identifying electronic data.

Personnel - Part III, C.1.c. (21 CFR 58.29)

Determine the following:

- Who was involved in the design, development, and validation of the computer system?
- Who is responsible for the operation of the computer system, including inputs, processing, and output of data?
- If computer system personnel have training commensurate with their responsibilities, including professional training and training in GLPs.
- Whether some computer system personnel are contractors who are present on-site full-time, or nearly full-time. The investigation should include these contractors as though they were employees of the firm. Specific inquiry may be needed to identify these contractors, as they may not appear on organization charts.

QAU Operations - Part III, C.2 (21 CFR 58.35(b-d))

- **Verify** SOPs exist and are being followed for QAU inspections of computer operations.

Facilities - Part III, C.3 (21 CFR 58.41 - 51)

- **Determine** that computerized operations and archived computer data are housed under appropriate environmental conditions.

Equipment - Part III, C.4 (21 CFR 58.61 - 63)

For computer systems, check that the following procedures exist and are documented:

- Validation study, including validation plan and documentation of the plan's completion.
- Maintenance of equipment, including storage capacity and back-up procedures.
- Control measures over changes made to the computer system, which include the evaluation of the change, necessary test design, test data, and final acceptance of the change.
- Evaluation of test data to assure that data is accurately transmitted and handled properly when analytical equipment is directly interfaced to the computer. and
- Procedures for emergency back-up of the computer system, (e.g., back-up battery system and data forms for recording data in case of a computer failure or power outage).

Testing Facility Operations - Part III, C.5 (21 CFR 58.81)

- **Verify** that a historical file of outdated or modified computer programs is maintained.

Records and Reports (21 CFR 58.185 - 195) (PART III C.10.b.)

- **Verify** that the final report contains the required elements in 58.185(a)(1-14), including a description of any computer program changes.

Storage and Retrieval of Records and Data - Part III, C.10.c. (21 CFR 58.190)

- Assess archive facilities for degree of controlled access and adequacy of environmental controls with respect to computer media storage conditions.
- **Determine** how and where computer data and backup copies are stored, that records are indexed in a way to allow access to data stored on electronic media, and that environmental conditions minimize deterioration.
- **Determine** how and where original computer data and backup copies are stored.

Hypertext updated April 3, 2001 by tmc

Navigational Assist

ORA OFFICE OF REGULATORY AFFAIRS FDA HOME PAGE INDEX SEARCH COMMENTS

Link to: Compliance References Focus Page | Page Top | Text Top

Links to: [ORA Home | FDA Home Page | FDA Site Index | FDA ORA Search Page | FDA Reader Comments]

FDA Investigations Operations Manual 2007

Chapter 5 Excerpt Related to Electronic Records Inspection

5.3.8.3 - Filmed or Electronic Records

When attempting to obtain records, you may find they are stored on microfilm, microfiche, or some form of a computerized management information system as electronic records.

5.3.8.3.1 - Microfilm/Microfiche and Electronic Information

You may encounter records stored on microfilm/microfiche or as electronic records on a computer system. Hard copy records obtained during the course of the inspection from these sources are handled the same as any hard copied records following procedures outline in IOM 5.3.8, 5.3.7.1 and 5.3.8.2.

NOTE: See CPG Section 130.400 for Agency Policy concerning microfilm and/or microfiche records. 21 CFR Part 11 contains information concerning Electronic Records and Electronic Signatures and may be of value to you.

5.3.8.3.2 - Electronic Information Received on CD-R, or other Electronic Storage Media

You may obtain electronic information, databases, or summary data from a firm's databases during an establishment inspection. The methods used must maintain the integrity of the electronic data and prevent unauthorized changes. Do not personally access a firm’s electronic records, databases, or source/raw data during the course of an inspection.

When it is necessary to access a firm's data during an inspection:

1. Oversee the firm's personnel accessing their system and have them answer your questions.
2. Request the firm run queries specific to the information of interest.
3. Have the firm generate reports/data to be copied to a CD or other electronic storage media, which you can subsequently analyze, or have the data printed in hardcopy.

Electronic data, such as blood bank databases, drug production records, medical device complaints, service records, returned products and other records are often dynamic data files with real time updating. Information from these files is generally provided at the time of the inspection. Your request may require the firm to develop one or more custom queries to provide the requested information. You must assume the query logic is not validated and take appropriate action to ensure the data is accurate and no data has been accidentally omitted due to a programming logic error occurring at the firm.

When appropriate, a copy of electronic data can be obtained on one or more CD-R, or other electronic storage media. If you provide the diskettes to the firm, use only new, previously unused and preformatted diskettes. An additional safeguard is to request the firm reformat the disk on their own computer to assure it is usable and "clean".

Any request for electronic information on a CD-R, or other electronic storage media must be made with a computer application in mind and the data obtained must be useful. Request for electronic information should be in a format compatible with software applications knowledgeable to you and available from the Agency. Converting files into different file formats is difficult and should not be attempted without the necessary knowledge and availability of conversion type programs where applicable. If help is needed for file conversion, assistance may be available within the district, region or from DFI HFC-130.

Any CD-R or other electronic storage media containing electronic information received during the course of an inspection should be considered and handled as master copies. The firm may or may not retain a copy of the information provided during the course of an inspection. Ask the individual providing the copy(s) to provide actual CD-R or other electronic storage media labeling information, such as filename(s), date and other information to facilitate their later identification of the CD-R or other electronic storage media and the data provided on the CD-R or other electronic storage media. The name of the appropriate software and version used to ensure readability of the information should also be maintained with the copy of the electronic information.

You should perform a virus scan of the master CD-R or other electronic storage media according to Agency requirements. Each master diskette should be write-protected, labeled and identified as you would any hard copy document.

There are no guarantees the files provided on CD-R or other electronic storage media will be useable data. It is your responsibility to make a working copy of each master CD-R or other electronic storage media. Before making any working copies from the master CD-R or other electronic storage media, confirmation should be made that the write-protection has been activated on each master diskette. You will need to use a computer to view the copied files and verify each file contains the information requested and the information is useable to you. Some electronic data files may be too large to open from a CD-R or other electronic storage media and must be loaded on a hard disk before opening. If this is the case, the file should be put on a subdirectory before opening and viewing.

As a general practice, any findings developed from electronic information provided by the firm should be requested in a hard copy format. The hard copy provided by the firm should then be used as an exhibit to support the investigator's observation. This will preclude or limit any errors that may have occurred from the investigator querying of the electronic information.

The master CD-R, diskettes or other electronic storage media, should be secured to assure the integrity of the data when used in a subsequent enforcement action. Identify the master copy as an exhibit, write-protect diskettes, and place in a suitable container, e.g., FDA-525, and officially seal. Mark the FDA-525 or other container as containing diskettes and to "Protect from magnetic fields." The diskette(s) should be stored as part of the exhibits with the original EIR. See IOM 5.10.5.1.

5.3.8.4 - Requesting and Working with Computerized Complaint and Failure Data

The auditing of FDA regulated firms has found that an increasing number of firms are developing and maintaining computerized complaint and failure data to meet GMP record requirements. Records, hardcopy and electronic, are becoming increasingly voluminous. The auditing of information contained in computerized databases is generally most effectively accomplished with the use of a computer.

Computer auditing of computerized complaints and failure data may require the transfer of electronic data to CD-R or other electronic storage media for you to use in your computer. You should use a computer and application software familiar to you to query information obtained in electronic format. You should not use the audited firm's equipment or personnel to perform repetitive queries or manipulation of the audited firm's own computerized data.

5.3.8.4.1 - Computerized Complaint and Failure Data

Requesting and obtaining electronic data on CD-R or other electronic storage media is becoming more common during the course of routine inspections. Providing computerized data on electronic media is advantageous to both you and the firm and can result in shorter inspection time. These types of databases contain large numbers of records, which can be easily and quickly queried if they are in electronic format. Inspection time would be lengthened if all such information was only provided in hardcopy format. It may result in you reentering all of the hardcopy data into a new database or reviewing volumes of documents. Be aware if the firm should generate custom software to provide requested electronic records, it would be difficult for you to validate or verify the firm's algorithm used to extract the requested data and ensure that records were not accidentally or deliberately omitted due to programming logic errors, data entry errors, etc.

5.3.8.4.2 - Requesting Computerized Data

Before requesting a copy of computerized data, you should determine several things including information about the size and contents of the database, the program used by the firm, and the program you will use, among others. The following steps are useful in preparing for an electronic record request.

1. Determine the firm's application program used to maintain the data of interest. This may be in a DOS compatible application program such as Access, Excel, Dbase, Paradox, Lotus 123 or others. It is best to obtain data files in a format compatible with application programs you will be using. Large data files with record counts in excess of 10,000 records are best converted to file formats that can be used by programs designed to handle such large databases. There are spreadsheet record limits in some commercial programs that would not allow these application programs to handle much over 5,000 records. Check the program you plan to use to ensure it can handle the file size you will be using.
2. Most large and real-time data files reside in mainframe or network systems requiring programming and downloading to a PC using an [Structured Query Language (SQL)] SQL format. Although data may be captured and downloaded in an SQL format, not all spreadsheet or database application software can load an SQL file. In addition, it may be difficult or impossible to manipulate data in that format. Problems can also be encountered downloading data from Apple computers to an IBM format. Successful conversions are possible if the firm selects the proper conversion format or you have conversion software designed to convert from an Apple to an IBM platform.
3. You may need to request an ASCII (American Standard Code for Information Interchange) text/flat file format. ASCII format is an industry standard, which assigns a unique code to every printable, keyboard, and screen character. An ASCII file should be stripped of all non [-] standard codes that are used by specific application programs for fonts, underlining, tabs, etc. The ASCII text file can be imported by all application programs, and once imported, can be restructured for the specific application program. ASCII delimited is the format of choice, with ASCII fixed length as an alternative. Care must be exercised in

specifying a hard carriage return at the end of each line to be DOS compatible, or additional conversion may be necessary before the file is useable.

4. You should determine what fields of information are routinely captured by the firm. This can be accomplished by requesting a printout of the data structure of the data file or observing the inputting of data at a computer terminal or workstation. It is common for databases to contain numbers or other coded information requiring translations from look up tables to give meaningful text. You should determine if information fields contain coded data, and if so, a code breakdown should be obtained. Information about code breakdowns should be located in the SOPs for that computerized system. Also be aware in relational databases, there may be linking data fields that exist in other tables that should also be considered in the overall data request.
5. If the files are too large to fit on a disk, file compression must be used. If possible, ask that the firm prepare the data in a compression format that is self-extracting. Self-extracting files are executable files and should be virus scanned before and after executing. All CD-R, diskettes or other electronic storage media should be scanned prior to being used on any FDA computer. Whatever compression utility is used, make sure you have the software to manipulate the files as needed.
6. You should always get the total record count of the data file provided by the firm. This count should be verified any time the file is loaded, converted, manipulated, or queried.

5.3.8.4.3 - Identification and Security of CD-R, Diskettes or Other Electronic Storage Media

You should follow these steps to ensure proper identification and security of CD-R or other electronic storage media:

1. Label each CD-R or other electronic storage media
    1. Firm name
    2. Date and your initials
    3. Initials by a representative of the firm (optional) If you provide the diskettes to be used, use only new and preformatted diskettes from an unopened box.
    4. The name of the appropriate software and version to ensure readability of the information
2. Make a working copy of CD-R or other electronic storage media
    1. Write protect the original diskette
    2. Virus scan the original diskette
    3. Copy the original CD-R or other electronic storage media

The original CD-R or other electronic storage media should not be used for manipulating data so as to maintain the integrity of the CD-R or other electronic storage media and data. NOTE: If a virus is detected, do not remove the virus from the source diskette provided by the firm. This may become evidence if it is suspected that the firm intentionally transferred the virus. Attempt to obtain another, uninfected copy of the data file from the firm.

Create a subdirectory on the computer hard drive:

1. Transfer data from the virus-free, working copy of the CD-R or other electronic storage media to your hard drive.
2. Virus scan any decompressed files before and after decompression. (Some virus scan software will scan compressed files but it is safer to scan all foreign files
3. You have now transferred confidential information to the hard drive and that information must be protected.
4. Upon completion of the use of the data, the file must be deleted and totally overwritten with a utility to wipe the data from the hard drive. A delete file operation is not adequate to totally remove the data from the hard drive.
5. Do not leave confidential files in any shared directories or e-mail.

5.3.8.4.4 - Data Integrity of Records Provided by Firm

Many manufacturers are using computers to store records concerning complaints, failure data, returned goods, servicing, testing results and others. Record traceability and data integrity are always concerns when you copy or use computerized data.

1. It is difficult to determine what records are to be designated as originals or copies of original records. It is important, when obtaining hardcopy or copy of computerized data, for you to capture some method of dating. The date of an electronic file can be captured by recording the date and time from a file listing in DOS or with File Manager in Windows. This may not always be possible, but some attempt should be made to date and time stamp electronic data.
2. Requests for most information from manufacturers will require the use of some custom software routine to generate the Investigator's requested information. Any data generated at the request of an Investigator should always be considered custom data. The firm will seldom validate or verify software routines used to generate data in response to your request. You should request a copy of any software program or scripts used to generate the computerized data provided. The request for the software program is not a request for a copy of the application program but a request for the special commands or programs created within the application program for the querying and extraction of data into a new data file. You should review the command structure to ensure it includes all data related to your request.

5.3.8.4.5 - Electronic Information for Official Documentation

During your use of queried data, if you find a violative situation, you should request the firm prepare a hardcopy report of the specific data that depicts the situation. (Do not request an entire copy of the data base and do not rely on the digital database or your extractions from the data to serve as official documentation.) Any records of interest, such as complaints, failure information, etc., noted from querying the computerized data should be copied from original hardcopy documents to support the findings in the database. You should also maintain the procedures or commands you used to find the violative situations in the data base. Follow procedures in IOM 5.3.8.3 for maintaining and identifying original disks.

5.3.8.5 - Listing of Records

If management requests a list of the copies of records you obtain, prepare it in duplicate and leave the original with the firm. Many firms prepare duplicate copies of documents requested during our inspections. In the interests of conserving inspectional time, you may ask the firm to prepare the list of copies concurrently with the photocopying and you then verify the accuracy. Do not use form FDA-484, Receipt for Samples. Describe the circumstances in your report including the name and title of the individual to whom you gave the list. Submit the duplicate list with your report as an Exhibit.

5.3.8.6 - Patient and/or Consumer Identification on Records

During the course of many types of inspections and investigations you will review and collect records which specifically identify (by name) patients or consumers. Under most state Privacy Laws this information is confidential. Some firms we inspect may mistakenly believe this information is not releasable to the federal government. However, Federal laws preempt State laws; with few exceptions we are entitled to review and copy the complete record, including the identifying patient/consumer names. The Agency is then required to maintain the confidentiality of the records/files, as with any confidential record you collect. Any disclosure of the information contained in the record(s) can only be by Law, i.e., judge's order, disclosure, Congressional order, etc.

General, routine guidance is as follows:

1. For records copied as a result of injury or complaint investigation, where you obtain patient identification, the identification should remain intact and stored in the official FDA files. Frequently, medical releases must be obtained from a complainant, consumer or "next-of-kin". At least one or two extra should be obtained and stored in the files.
2. For methadone inspections, continue the Agency policy of deleting patient identification specific to the patient (name, SSN, Driver License #, etc.).
3. For any inspection/investigation involving a regulation required Informed Consent, such as clinical investigations, IRBs, bioequivalence testing, etc., patient identification should remain intact and stored in the official FDA files.
4. For most others, such as MQSA, plasmapheresis, blood donations, etc., only the patient initials and unique identifier supplied by the firm (such as donor number, donation number, etc.) need be routinely retained in the FDA files.

It is not uncommon for a firm to voluntarily purge the documents of the pertinent identifiers as they are copied. You must verify (by direct comparison to the original document) you received an accurate reproduction of the original, minus the agreed to purging, prior to accepting the copy.

As with any inspection there are times when the specific identifiers must be obtained, copied and retained, such as if/when further interview of the patient/consumer could be necessary. If in doubt, obtain the data. It is always easier to delete later than to return to obtain the information, especially in the few cases where questionable practices may result in the loss of the information.

All documents obtained containing confidential identifiers will be maintained as all documents obtained by FDA containing confidential information, i.e., in the official FDA files. Confidential identifiers may be flagged in the official FDA files for reference by reviewers to assure no confidential data are released under FOIA

CHAPTER 22C

TAB K

INSPECTION OF COMPUTER SYSTEMS

OBJECTIVES

After this chapter, you will:

- Understand how the GMPs are applied to computer systems
- Know the similarities and differences between computer systems and other systems
- Comprehend what is meant by validation of computer systems
- Know one method for conducting an inspection of a computer system

REFERENCES

1. Title 21, Code of Federal Regulations, Parts 210, 211, 606, 830
2. Guide to Inspection of Computerized Systems in Drug Processing, Reference Materials and Training Aids for Investigators, U.S. Food & Drug Administration, February 1983
3. Software Development Activities, Reference Materials and Training Aids for Investigators, U.S. Food & Drug Administration, July 1987
4. Medical Device Good Manufacturing Practices Manual, Fifth Edition, U.S. Government Printing Office, 1991
5. FDA-2609 Blood Bank Inspection Checklist and Report, Section K, and Instruction Booklet, U.S. Food & Drug Administration
6. CBER Memo, Requirements for Computerization of Blood Establishments, September 8, 1989
7. CBER Memo, Recommendations for Implementation of Computerization in Blood Establishments, April 6, 1988
8. Guideline on the General Principles of Process Validation, U.S. Food and Drug Administration, CDER, CDRH, May 1987

CHAPTER OUTLINE

HIGH LEVEL SPECIFICATION REVIEW

- Do stated goals and objectives for software remain consistent with system goals and objectives?
- Have important interfaces to all system elements been described?
- Is information flow and structure adequately defined for the problem domain?
- Are diagrams clear? Can each stand alone without supplementary text?
- Do major functions remain within scope and has each been adequately described?
- Is the behavior of the software consistent with the information it must process and the functions it must perform?
- Are design constraints realistic?
- What is the technological risk of development?
- Have alternative software requirements been considered?
- Do inconsistencies, omissions, or redundancy exist?
- Has the user reviewed the *Preliminary User's Manual* or prototype?

DETAILED SPECIFICATION REVIEW

- Be on the lookout for persuasive connectors, e.g., "certainly", "therefore", "clearly", "obviously", "it follows that", and ask, "Why are they present?"
- Watch out for vague terms, e.g., "some", "sometimes", "often", "usually", "ordinarily", "most", "mostly", and ask for clarification.
- When lists are given, but not completed, be sure all terms are understood. Keys to look for: "etc.", "and so forth", "and so on", "such as".
- Be sure ranges don't contain unstated assumptions, e.g., "Valid codes range from 10 to 100." Integer? Real? Hex?
- Beware of vague verbs such as; "handled", "rejected", "processed", "skipped", "eliminated". There are many ways they can be interpreted.
- Beware of "dangling" pronouns; e.g., "The I/O module communicates with the data validation module and *its* control flag is set." Whose control flag?
- Look for statements that imply certainty; e.g., "always", "every", "all", "none", "never". Then ask for proof.
- When a term is explicitly defined in one place, try substituting the definition for other occurrences of the term.
- When a structure is described in words, draw a picture to aid in understanding.
- When a calculation is specified, work at least two examples.

from Software Engineering, A Practitioner's Approach, 3rd Ed. by R.S. Pressman, McGraw-Hill Inc.

I. Definitions

A. Validation - Establishing documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its pre-determined specifications and quality attributes.

B. Computer System - any system which includes a central processing unit, such as a microprocessor, and the software which determines the behavior of that system, including software installed in programmable memory chips.

C. Hardware - the mechanical components of the computer system such as the processor, displays, input and output devices, other peripheral devices, etc.

D. Software - the logical statements which instruct the computer to perform required actions. The software component also includes software documentation which is an essential part of software.

E. Sensors - devices for measuring data (analog) from outside world.

F. Actuators - devices which accept computer signals and convert those signals to some action on the outside world.

II. Computer Systems and the GMPs

A. Primary uses of computers in regulated industry

1. Process control - all industries

2. Process monitoring - all industries

3. Manufacturing and quality control records management - all industries

4. Device component - medical devices

5. Medical device - medical devices and blood banks

B. Hardware can be considered as equipment in a manufacturing process or as a medical device component

1. Considered as equipment, must comply with equipment sections of respective GMPs

a. Must be installed in accordance with the manufacturer's instructions

b. Sensors and actuators may require calibration

c. Must be maintained in accordance with SOPs

2. Considered as a device component, must comply with component section of device GMPs

C. Software can be considered as:

Records or SOPs when used in a manufacturing process or components when used in a medical device or as a device itself

1. Automation is essentially replacing written records and manual procedures with computer software for data recording

2. Considered as records, software must preserve record integrity

a. Data input must be accurate

b. Data output must be accurate

c. Data must be secure from unauthorized changes

d. Data changes must be documented

3. Considered as SOPs, software is a manufacturing process and must be validated

4. Considered as a device component, software must be shown to perform its intended function according to specifications and must not perform unintended functions

III. Computer System Validation

A. Validation of computer systems, as with any system, must be appropriate for the intended use

1. The more complex the system, the more rigorous the validation

2. The more critical the system function, the more detailed the required validation

B. Intended function

1. The system specifications or requirements define the intended function

2. The intended function must be defined in sufficient detail and in unambiguous language

a. The system developer must be able to understand what is required without assuming what was intended

b. The user must be able to understand the specification and determine that the system fulfills the specified need

3. The specification should describe

a. Valid system inputs and where those inputs come from

b. The data manipulations to be performed on the input data

c. The expected system outputs

d. System limitations and constraints

e. Any system defaults

f. All validity and error checking of system inputs and outputs

4. The intended function must be described with objective, measurable criteria

5. The specification must be modifiable through specified controlled procedures

C. Documented evidence

1. Software includes documentation

2. Includes specifications, design documents, source code, test documents, user's manuals, maintenance documents, etc.

3. Documents should cover the system being used

D. Accurate and reliable

1. Test documents should demonstrate accuracy

a. Functional test - compliance to specifications

b. Structural test - all of system tested

c. Worse-case testing

2. Reliability of computer systems difficult to demonstrate

a. A reliable system performs consistently from one run to the next and does not perform unintended functions

b. Errors occur due to improper handling of inputs

c. Improper input handling caused by

(1) Invalid data entry

(2) Sequence or combination of inputs not anticipated by the programmer

(3) Mistake in the logic of the program

(4) Using system in a manner not intended

d. Number of possible input combinations is astronomical

e. Careful design of test and repeat runs of test protocols are needed

IV. Computer System Inspection

A. Determine what computer systems are used

1. User may not be aware of all computers used in operations

2. Report all computers observed but choose one or two systems for detailed coverage

3. Some criteria for choosing system(s) for inspection

a. Computers involved in most critical operations

b. Systems where unexpected data input is most likely to occur - such as heavy manual data input from keyboard

c. Review error/problem reports and complaints for involvement of computers

d. Review computer maintenance and change control records, especially for software revisions and "bug" fixes

4. Compare user's explanation of system function against user's manuals, SOPs, and system observation

B. Computer Hardware

1. Determine if the computer was installed according to manufacturer's specifications

2. Determine what types of sensors (if any) feed data to the computer

   a. Check calibration of sensors

   b. Check maintenance of sensor equipment

   c. Check procedures, manual and software, for assuring validity of input data

3. Determine what types of actuators (if any) accomplish actions controlled by the computer system.

   a. Check calibration of actuators

   b. Check maintenance of actuators

   c. Determine how the computer verifies completion of action and current state of actuator

   d. Check for any diagrams of how hardware is connected in system

   e. Check for performance of any routine maintenance required by computer manufacturer

C. Computer Software

1. Determine what programs are used and what version is being used

2. Note who developed the software

3. Review the written specifications for the software

   a. Compare the specifications against the software in use to determine if specs describe current system

   b. Determine if the requirements are clear and measurable

   c. See if you can define test cases which demonstrate compliance of the software with the specifications

   d. Check for descriptions of type of input data and validity checking of that data

e. Check for descriptions of manipulations which the program will perform on input data and descriptions of what output is expected

f. Determine what the user thinks the system does if there are no written specifications and determine how the user knows this

4. Review the software test procedures and test documentation

a. Determine who performed the software testing and review the testing closely if only the programmer performed testing

b. Review the test documentation. Does it:

(1) Define the test data used

(2) Define exactly what is tested with the test data

(3) Describe the expected results from the test data

(4) Provide actual test results or just the test personnel's opinion (pass/fail type results)

(5) Compare the test records against the specifications to see if all specifications have been met

(6) Look for testing of boundary conditions, invalid inputs, and special cases such as 0, negative numbers, etc. (worse-case tests)

c. Determine whether the firm tested only against specifications or did they also develop test cases to assure all of the program was tested (structural testing)

5. Review the user's manual or SOPs for any programs used and determine if the user made any attempt to verify the accuracy of this documentation

6. Observe the computer system in use and compare your observations against the use(s) described in the documentation and/or verbal descriptions provided by the user

7. Determine if software is included in firm's change control procedures

    a. All software changes are design modifications

    b. Check for review and approval of software changes

    c. Determine if specifications were written for software changes

    d. Be alert for software which has had frequent modifications

    e. Review the testing of software modifications

D. Computer System Security

1. Review the firm's written procedures for computer system security

2. Determine what personnel have access to the system and what personnel are authorized to make changes to programs or data in system

3. Check for training of personnel in computer security

4. Be alert for electronic audit trails

5. Determine if there are written procedures for data and system recovery in the event of system failure or disasters

E. Training

1. Review training of system users

2. Determine if the firm has manual procedures in the event of system failure and if personnel are periodically trained in these manual procedures

V. Summary

A. Computer systems are very complex

B. A planned, methodical approach to inspection works just as with inspection of other systems

C. Validation of a computer system means demonstrating that the system performs the intended functions and does not perform unintended functions

D. Computers are regulated using the existing GMPs

✓REVIEW

1. What are three basic components of a computer system?

2. How are computer systems regulated under the GMPs?

3. What is meant by computer system validation?

4. How are computer systems similar to other types of systems? How are they different?

5. Name two considerations in choosing a computer system for in-depth inspection?

6. How does a firm demonstrate the accuracy and reliability of computer systems?

7. What should be covered during the inspection of a computer system?

SOFTWARE CPR®
CRISIS PREVENTION AND RECOVERY, LLC

www.softwarecpr.com
20 Berkshire Drive
Winchester, MA 01890
781-721-2921

Copyright

SoftwareCPR authorizes its clients and SoftwareCPR.com subscribers use of this document for internal review and training. Any other use or dissemination of this document is expressly prohibited without the written authorization of SoftwareCPR.

Legal Disclaimer

The training document example that follows should only be applied in the appropriate context with oversight by regulatory and software professionals with direct knowledge and experience with the topics presented. The document should not be used as a cookbook since it is not adequate for some situations and may be excessive for others.

While SoftwareCPR attempts to assure the accuracy of information presented no guarantees are made since regulatory interpretations and enforcement practices are constantly changing, and are not entirely uniform in their application.

Disclaimer of Warranties: The information is provided AS IS, without warranties of any kind. CPRLLC does not represent or warrant that any information or data provided herein is suitable for a particular purpose. CPRLLC hereby disclaims and negates any and all warranties, whether express or implied, relating to such information and data, including the warranties of merchantability and fitness for a particular purpose.

IT/Network Support Procedures Handbook Training Example

Aug 28, 2002

SoftwareCPR Note:

This training example focuses on aspects of regulatory concern for validation and Part 11 compliance. If a handbook (rather then individual procedures) approach is taken then it can be helpful to include additional sections containing information useful to administrators that may not be important for regulatory purposes.

This is not a template to assure compliance – it is a training example used for discussion and to trigger ideas useful for creating tailored procedures/handbooks for specific environments.

TABLE OF CONTENTS

1 General Information

1.1 IT Organization

1.2 Responsibilities

2 Network/Server Environment

2.1 Network Architecture

3 Problem Monitoring

3.1 IT Monitoring

3.2 User Problem Reporting

3.3 Documenting a Problem

Server and network hardware problems and errors should be documented in the Logbook in the main server room. Items in log should include equipment ID date, time down, time up, and action taken, by who.

4 Backups

4.1 Backup Schedule

4.1.1 Servers

4.1.1.1 <u>Full Backups</u>

4.1.1.1.1 <u>Weekly</u>

4.1.1.1.2 <u>Monthly</u>

4.1.1.1.3 <u>Offsite</u>

4.1.1.2 Daily Incrementals

4.1.1.3 Rotation Schedule

4.1.1.4 Media Type, longevity, and replacement

For long term backups and archives …

4.1.2 Routers, Firewalls, and other network equipment backups

Configuration and firmware backups for security related equipment is performed each time the firmware is updated for security purposes as well as monthly.

4.1.3 User Workstation/Client Backup

<<If IT provides network service for this>>

5 Archives

Users may request archive copies of specific applications and data for long term storage…

6 Recovery

<<This should include steps to take including notification of users responsible for regulated applications and verification of proper restoral of latest available backup.>>

6.1 Server

6.2 Application

6.3 Disaster

7 Security

7.1 Physical and Hardware

7.1.1 Computer room physical security

7.2 Internet

7.2.1 Firewall

7.2.2 FTP access

7.2.3 Secure Sockets Usage

7.2.4 VPN and Dialup

Encryption…

7.2.5 Workstation

7.2.6 Denial of Service

7.3 E-mail

7.3.1 Instant messaging

7.3.2 Spam

7.4 Data and System

7.4.1 Server Logins

<< Include or reference procedure, approval requirements, and documentation/forms to retain >>

7.4.2 Rights and Permissions Hierarchy/ User groups

7.5 Threat Response

7.6 User Accounts

7.6.1 User Network Activation Process

7.6.1.1 Activation

Fill out a New User Account Activation Form and get approvals prior to activation.
File the form for record retention.

7.6.1.2 Deactivation

Fill out a deactivation form.
File the form for record retention after deactivation

7.6.2 Uniqueness

Account IDs must be a minimum of nn characters with at least xx numbers or special characters.

No two account IDs can be identical.

Account IDs can not be reused even after deactivation.

7.7 Login Policies

7.7.1 *Password Restrictions*

Maximum Password Age: Expire in 90 Days
Minimum Password Length: At Least 6 Characters

7.7.2 *Account lockout*

Lockout after 6 bad logon attempts
Reset count after 10 minutes
Lockout duration 10 minutes

7.7.3 *Periodic Review*

At the start of each year access lists will be circulated to user line management. They will redline any corrections needed. These will be implemented and the redline copy filed and retained.

7.7.4 *Vendor Access*

7.7.5 *Electronic Signatures*

<< If e-sigs are used for regulated purposes include admin responsibilities associated with 21 CFR Part 11 such as verification of identity, forgery protections, policy on management requests that could compromise esig validity… >>

7.7.6 *System Documentation Security*

<< Access protection to secure information that could compromise security or record integrity >>

8 Virus Protection

8.1 Server

8.2 Workstation

8.3 Email

8.4 Virus File Updates

8.5 Virus Response

<< notifications, determination of impact on regulated data, inoculation or recovery…>>

9 Application/Database Configuration Management

9.1 Updates

9.1.1 Security Patches

9.1.2 Configuration changes

9.1.3 Application minor updates

9.1.4 Application Major updates

9.1.5 User notification

<< Advance notice to users with regulated applications (to allow for evaluation and re-validation) except for emergency security or recovery situations in which ASAP notification is allowable. >>

9.2 Database Administration

9.2.1 Purging

<< under what conditions is purging done and how is loss of regulated records or omission of such information from reports prevented. >>

9.2.2 Custom Queries

<< If not handled on application-specific basis governed by application-specific procedures include or reference controls and documented verification requirements >>.

9.2.3 Data Modification

<< If not handled on application-specific basis governed by application-specific procedures include or reference procedures and documented approvals and documented log of data changes made.>>

9.2.4 Configuration Changes

<< If not handled on application-specific basis governed by application-specific procedures include or reference procedures and documented verification of table/configuration changes made.>>

10 Training

10.1 IT Staff Training

10.1.1 Technical

10.1.2 Regulatory and Validation

10.1.3 Security

10.2 User Training

10.2.1 Security

<< good password practices, …

10.2.2 Problem Reporting

11 Special Application-specific Procedures

<< Describe any special support requirements and documentation required for specific applications. This could include for example any special archiving or security requirements or purging or error checking/reporting that is done only for specific applications. If IT is responsible for validation of specific applications or classes of application identify and reference validation procedures that apply. >>

5/12/2005 Not necessarily compliant - specifics to be defined for each company/product type.

Validation Planning Roadmap Example

Validation Assurance Level

Activity	A	B	C
Risk Assessment	Explanation of why defects would not affect final product or support or required regulatory records.	Explanation of why defects would not be likely to have a significant affect on final product or support or required regulatory records.	Analysis of specific functionality and its potential impact and identification of internal and external risk control measures
Plans	Not required if general validation procedures exist. If needed can be prepared and approved within the group responsible. No separate plan needed just include information in validation protocol.	Required prior to testing but may be approved within the group responsible.	Required upon completion of requirements, must be approved by corporate validation group or QA, and must be updated and reapproved for changes as project progresses.
Approvals	Responsible party only.	Local approval by a designated manager or QA person within the department responsible.	Approvals of plans, documents, and results by corporate validation group or corporate QA, and documents must be updated and reapproved for changes as project progresses.
Reviews	Discretionary and local.	Informal reviews required with only a log or note that they occurred.	Formal review of requirements and final tests and results.
Requirements	High level intended use statement and major features/functions	High level intended use statement and description of each workflow, key requirement, and key data/records for risk related functionality.	Detailed requirements specifications for risk related functionality.
Design	Statement of basic platform.	Statement of basic platform and identification of major components including versions of third party components.	Detailed platform, architecture, algorithm and data structures for custom components and detailed identification of third party components and all configuration options, macros, and tables for risk related modules and components.
Testing	Functional test cases mimicking intended use.	Test plan describing approach and types of testing to be performed and functional test cases for each workflow including testing of each key requirement, data, and error checking.	Workflow and functional testing as well as testing all algorithms and data under a range of conditions normal and abnormal and stress conditions with emphasis on aspects identified as critical in the risk assessment.
Test Evidence	Summary of results with objective evidence for intended use.	Summary of results with detailed objective evidence for each test case for risk related functionality and components.	Summary of results with detailed objective evidence for each test case.
Configuration Management	General corporate procedures may be sufficient or local procedures can be developed.	Specific work instructions for this application based on technology, tools and environment as well as group responsible.	Specific work instructions for this application based on technology, tools and environment as well as group responsible AND approval by corporate QA or Validation authority.
Security	General corporate procedures may be sufficient or local procedures can be developed.	Specific work instructions for this application based on technology, tools and environment as well as group responsilbe.	Specific work instructions for this application based on technology, tools and environment as well as group responsible AND approval by corporate QA or Validation authority.
Backups	General corporate procedures may be sufficient or local procedures can be developed.	Specific work instructions for this application based on technology, tools and environment as well as group responsible.	Specific work instructions for this application based on technology, tools and environment as well as group responsible AND approval by corporate QA or Validation authority..
Administration	General corporate procedures may be sufficient or local procedures can be developed.	Specific work instructions for this application based on technology, tools and environment as well as group responsilbe.	Specific work instructions for this application based on technology, tools and environment as well as group responsible AND approval by corporate QA or Validation authority.
User Training	General corporate procedures may be sufficient or local procedures can be developed.	Specific work instructions for this application based on technology, tools and environment as well as group responsilbe.	Specific work instructions for this application based on technology, tools and environment as well as group responsible AND approval by corporate QA or Validation authority.

Software Project Validation Data Sheet

Date:

Project Name:		**Project Leader:**	
Usage:	Mfg Product	**Complexity:**	High Medium Low
Type:	Off-the-shelf Custom Contract	**Criticality:**	High Medium Low
Status:	In Development Released		
Deliverable Names(s):			
Masters' Location:			
Records Location(s):			

Comments:

Application Name: **Date:**

1. Software Requirements (Functionality & purpose including security and performance features and, if applicable, hazard analysis):

Intended use summary. If simple include requirements. If complex reference separate spec.Could reference sections of SOPs and sections of Vendor documentation and additional internal specs.

2. Design Description (Approach taken to implementation, partitioning, key design decisions):

If OTSS include configuration options. If simple could be combined with 1 above.

3. Coding & Code Control (Coding standards, source code control, directory structure):

4. Testing (Consider functional, platform, negative, fault, security, stress, performance, usability, boundary testing , & independent test methods, requirements traece/coverage):

5. Change Control (Normal EC N process or describe the change request and change tracking mechanisms):

6. Release Control (Normal EC and rev numbering, change summary, approvals, media control, installation, removal of old revs):

7. Backup & Archival:

8. Tools (Compilers, build tools, packages, their version numbers, tools version control, etc.):

9. HW & SW Environment (HW requirements, software packages & their versions, ESD, dust,etc.*):*

10. User Training & Documentation (User and administrator requirements & approach to training):

11. Monitoring (Defect reporting and analysis, periodic checks on current version, actual performance security, ,etc.):

..

12. Other Relevant SOPs and/or SQA Plan

13. Other: e.g., if OTSS then Vendor/OTSS qualification or vendor provided validation information

SOFTWARECPR
CRISIS PREVENTION AND RECOVERY, LLC

www.softwarecpr.com
20 Berkshire Drive
Winchester, MA 01890
781-721-2921

Copyright

Legal Disclaimer

The training document example that follows should only be applied in the appropriate context with oversight by regulatory and software professionals with direct knowledge and experience with the topics presented. The document should not be used as a cookbook since it is not adequate for some situations and may be excessive for others.

While SoftwareCPR attempts to assure the accuracy of information presented no guarantees are made since regulatory interpretations and enforcement practices are constantly changing, and are not entirely uniform in their application.

Disclaimer of Warranties: The information is provided AS IS, without warranties of any kind. CPRLLC does not represent or warrant that any information or data provided herein is suitable for a particular purpose. CPRLLC hereby disclaims and negates any and all warranties, whether express or implied, relating to such information and data, including the warranties of merchantability and fitness for a particular purpose.

OTS Spreadsheet Software Checklist

Date:____________ Location:____________________

Process:____

Application Software

Manufacturer: ______________________

Title: ______________________ Version:__________

Computer System Requirements

Single□ Multi□ Network□

Hardware CPU 386□ 486□ Pentium□

RAM Min________ Max________

Software OS Dos□ Windows 3.1□
Windows95□ Windows NT□

Drivers ______________________

Utilities ______________________

Background SW ______________________

Storage Local □ Network □ Both □

Hard Disk □L N Floppies □L N

Mag/Opt □L N Other □_________L N

Comments____
____
____
____

OTS Spreadsheet Software Checklist

Security

Network ____________________

OTS ____________________

Master Copy ____________________

Backup ____________________

Comments____

Spreadsheet Specifics

Title: ____

Revision: ______________

Revision date: ______________

HW Residence: ______________

SOP: ____

Products affected: ____

Primary use: ____________________________

Distribution: ____________________________

	yes	no		yes	no
Passwords	□	□	Alerts	□	□
Error Control	□	□	User Modifiable	□	□
Macros	□	□	Functions	□	□
Sig. Figure Control	□	□	Cell Constraints	□	□

Comments: ____

____

____

____

______________ Reviewed By

______________ Date