code of hilations

Computer System Validation Training

Presented by:



Focus Compliance & Validation Services



ruary 9, 2010

Agenda (1 of 2)

- Introduction
 - What Is Computer System Validation (CSV)?
 - What Is the Rationale?
 - Who Is the Audience?
 - What Types of Hardware/Software Are Regulated by the FDA?
 - Where Is Computer Applicable?
- FDA CFR and Guidance Documentation
 - Some Key Terms and Definitions
 - Requirements
 - Specifications
 - System Design
 - Software Is Different From Hardware

@ 2010 Focus Compliance & Validation Services and SyberWorks, Inc.



Agenda (2 of 2)

- Definitions and Terminology
- Approach to Validation



Poll 1

2010 Focus Compliance & Validation Services and SyberWorks, Inc.





Myth #1

Validation is a project related activity
Once completed it can be largely ignored





Truth

- Validation is a journey, not a destination
- Validation is best performed when it is practiced in a life-cycle model, using a <u>cradle-to-grave</u> approach
 - This provides maximum benefit in compliance and finance

What is Validation?

• Validation, Process (FDA)

 <u>Establishing documented evidence</u>, which provides a <u>high degree of</u> <u>assurance</u> that a specific process will <u>consistently</u> produce a product <u>meeting</u> its predetermined <u>specifications</u>

Validation, Software (FDA)

 <u>Confirmation by examination</u> and provision of <u>objective evidence</u> that <u>software specifications conform</u> to user needs and intended uses, and that the particular <u>requirements</u> implemented through the software can be <u>consistently fulfilled</u>

Validation, Software (NBS)

- <u>Determination of the correctness</u> of the final program or <u>software</u> produced from a development project <u>with respect to</u> the user needs and <u>requirements</u>
- Validation is usually accomplished by verifying each stage of the software development life cycle



What Is The Rationale For CSV?

- The Quality System Regulation, last revision in the Federal Register October 7, 1996 and made effective June 1, 1997, established Software Validation as a requirement
 - Title 21 Code of Federal Regulations (CFR) Part 820
 - Section 820.05 Quality System:
 - Each manufacturer shall establish and maintain a <u>quality</u> <u>system</u> that is appropriate for the specific medical device(s) designed or manufactured, and that meets the requirements of this part
 - Section 820.70i Production and Process Controls:
 - Automated processes. When computers or automated data processing systems are used as part of production or the quality system, the manufacturer shall validate computer software for its intended use <u>according to an established</u> <u>protocol</u>. All software changes shall be validated before approval and issuance. These validation activities and results shall be documented



What Is The Rationale For CSV?

- The intent for the Requirement of CSV is to:
 - <u>Ensure</u> that finished <u>product(s) will be safe and effective</u> and otherwise in <u>compliance</u> with The Federal Food, Drug, and Cosmetic Act
 - <u>Establish</u> the <u>basic requirements</u> applicable to manufacturers of finished medical devices and finished pharmaceuticals





2010 Focus Compliance & Validation Services and SyberWorks, Inc.





Myth #2

• Validation can be accomplished by a small cadre of individuals with minimal imposition on the rest of the organization



Truth

- Validation is everyone's job
- Each operational unit within an organization must contribute to the overall effort to ensure success
- No single operational unit can hope to satisfy <u>all</u> validation requirements and reap the benefits alone

Who Is The Audience?

- The General Principles of Software Validation Guide States:
 - Persons subject to the medical device quality system regulation
 - Persons responsible for design, development, or production of medical device software
 - Persons responsible for design, development, production, or procurement of automated tools used for 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

What Types of Computer Systems Are Regulated By The FDA? Regulatory Governey





Process Validation





Design Validation Risk Analysis

Device Software



Quality System Software Design Tool Software Testing Software CAPA Software **Validation**

@ 2010 Focus Compliance & Validation Services and SyberWorks, Inc



Computer System Projects Take Many Forms

- MS Excel[™] spreadsheet or Minitab[™] project
- Quality System including CAPA tracking system, document management, or training
- Software that is a medical device
- Software embedded in a medical device
- Controls software for the production of active pharmaceutical ingredient, biologic or medical device
- Laboratory Information Management System (LIMs)
- Automated Testing systems

Definitions and Terminology

- Establish (21CFR 820.3(k))
 - is defined to mean define, document, and implement

• Requirement (IEEE)

- (1) A <u>condition or capability</u> needed by a user to solve a problem or achieve an objective
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents
- (3) A documented representation of a condition or capability as in (1) or (2)

• Specification (IEEE)

 A <u>document</u> that specifies, in a complete, precise, verifiable manner, the requirements, design, behavior, or other characteristics of a system or component, and often, the procedures for determining whether these provisions have been satisfied.

• Traceability (IEEE)

- (1) The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor or master – subordinate relationship to one another
- (2) The degree to which each element in a software development product establishes it reason for existing.

• **Traceability Analysis** (IEEE) The tracing of:

- (1) System Requirements Specification requirements to system requirements in concept documentation
- (2) software design descriptions to software requirements specification and software requirement specifications to software design description
- (3)source code to corresponding design specifications and design specification to source code.

- Traceability Matrix (IEEE)
 - a matrix that records the relationship between two or more products
- **V&V** Verification and Validation Separate and distinct
- Verification (FDA Guidance)
 - provides objective evidence that the design outputs of a particular phase of the life cycle meet all of the specified requirements for that phase.

• VV&T (NIST) Validation, Verification, & Testing

 Used as an entity to define a procedure of review, analysis, and testing throughout the system life cycle to discover errors, determine functionality, and ensure the production of quality software

Validation (FDA)

- Establishing documented objective evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specification and quality attributes
- Hazard (IEC 300-3)
 - Source of potential harm or a situation with a potential for harm.
- Risk Analysis (IEC 300-3)
 - Systematic use of available information to identify hazards and to estimate the risk to individuals or populations, property or the environment.

- Validation Protocol (FDA) A written plan stating how validation will be conducted, including test parameters, product characteristics, production equipment, and decision points on what constitutes acceptable test results.
- Software Validation (FDA Guidance) Determination of the correctness of the final program or software produced from a development project with respect to the user needs and requirements. Validation is usually accomplished by verifying each stage of the software development life cycle.

• Software Verification – (FDA Guidance) In general the demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle.

Benefits of Software Validation

- Software Validation is a critical tool for:
 - Assuring quality of software and software automated operations
 - Increasing usability and reliability of a finished product
 - Decreasing failure rates, which in turn result in fewer recalls and corrective actions
 - Lessening risk to patients and users
 - Reducing liability to manufacturers by providing defensibility for how they carried-out software validation
 - Reducing long-term costs



Poll 3

2010 Focus Compliance & Validation Services and SyberWorks, Inc.

Approach to Validation Whether configuring COTS or Writing code

- Project Planning
- Design Requirements and Specifications
- Hazard Identification, Risk Analysis, and Mitigation
- Validation
 - Hardware and Software
 - IQ/OQ/PQ
- Repeat as necessary
 - Changes
 - Upgrades



What is the Computer System

- Software validation <u>must be considered</u> within the context of the overall design validation for the software product
 - FDA expectation is that software is part of a computerized system where the system is the hardware, the software, and the associated procedures on how to use and maintain the system
- The documented requirements specification represents the user's needs and intended uses from which the software product is developed
- A primary goal of software validation is to demonstrate that all completed software products comply with all documented software requirements. This includes:
 - Confirmation of <u>Conformance</u> to <u>All Software Specifications</u>
 - Confirmation of *Traceability* of *All Requirements to Specifications*

Software Is Different From Hardware

- While software shares many of the same engineering tasks as hardware, it has some very important differences. For example:
 - Software problems are traceable to errors made during the design and development processes
 - Software is complex due to its branching capabilities
 - Testing alone cannot fully verify that software is complete and correct
 - Software does not wear out like hardware but it is enhanced by finding and eliminating defects
 - Software failures occur without advanced warning because of hidden and latent defects
 - Software can be quickly and easily changed unknowingly without strict engineering control

• In summary

 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. This is true regardless whether software is standalone or part of an embedded hardware device

Principles of Software Validation

- Requirements
- + Defect Prevention
- Time and Effort
- Software Life Cycle
- Plans
- + Procedures
- + Software Validation After a Change
- + Validation Coverage
- + Independence of Review
- + Flexibility, and Responsibility
- + Retirement

Poll 4

2010 Focus Compliance & Validation Services and SyberWorks, Inc.

• Purpose

 A high-level definition of the product or portion of the product whose requirements are specified in the document

Scope

A brief narrative defining the extent of the document

• Definitions

 A list of each defined term or acronym referenced within the document

• Overall Description and Product Perspective

 A description of the origin and background for the product being supported by the specified system

General Requirements

- A brief narrative introducing the section
- A list of each general requirement and its description, which includes, but is not limited to: features, functions, hardware, operating system, ancillary software, usability, reliability, maintainability, regulatory, safety, and performance

Assumptions

- A brief narrative introducing the section
- A list of assumptions known about the specific product and/or computer system

• Risks

- A brief narrative introducing the section
- A list of each risk that is known to be associated with the specified product and/or computer system, which includes the rationale for the risk

Project Boundaries

- A brief narrative introducing the section
- A list of each Project Boundary that limits the specified product and/or Computer System Development effort, which includes its explanation

• Project Timetable and Milestones

- A brief narrative introducing the section
- A list of each major project milestone, including start of project milestone, any significant intermediate milestone, and the end of the project milestone
- A schedule (created using a software tool such as MS Project) showing all tasks, their titles, start and stop dates and times, durations, predecessors, successors, assigned resources, percentage complete, and comments

Resources

- A brief narrative introducing the section
- A list of each resource necessary to successfully complete the specified product and/or Computer System Development project, including resource types of capital, human, and materials

Roles and Responsibilities

- A brief narrative introducing the section
- A list of each role that is necessary to successfully complete the specified product and/or Computer System Development project and its responsibilities
- An organizational chart

• Accompanying Documentation

- A brief narrative introducing the section
- A list of each document that is known to be needed, to be written, and/or to accompany this product and/or system <u>in addition to the System Development</u> <u>Life Cycle (SDLC) documentation</u>

References

- A list of any other document and/or source material referenced within this document

• Approvals

 A table containing named roles responsible for approving this document and their signatures

Computer System Requirements Purpose

• The purpose is:

- Determining the scope of detail-level requirements for a computer system development project by leading stakeholders through a structured requirements-gathering process
- Providing stakeholders a document that easily reiterates their set of detail-level requirements
- Providing a detailed enough requirements document that precludes a programmer from making any ad hoc design decisions and enables that programmer to create test scenarios
- Providing documentation that serves as a portion of the objective evidence necessary for successful validation of a Computer System Development project

Computer System Requirements Outline

- Regulatory Requirement Title
 - A short but descriptive title that is usually five words or less

Brief Description

 A short but descriptive narrative that is usually two paragraphs or less describing the Regulatory Requirement

Assumptions

 If applicable, a list of assumptions that have been applied to the use of this Regulatory Requirement

Constraints

- If applicable, a list of all constraints that apply to the use of this Regulatory Requirement
- If none exist, mark "N/A"

@ 2010 Focus Compliance & Validation Services and SyberWorks, Inc

Computer System Requirements Outline

• Special Requirements

- If applicable, a list of special requirements unique to this Regulatory Requirement
- If none exist, mark "N/A"

Business Rules

- If applicable, a list of business rules unique to this Regulatory Requirement
- If none exist, mark "N/A"

Usability Requirements

- If applicable, a list of usability requirements unique to this Regulatory Requirement
- If none exist, mark "N/A"

• Approvals

 A table containing named-roles responsible for approving this document

@ 2010 Focus Compliance & Validation Services and SyberWorks, Inc.

Risk Analysis - Purpose

• The purpose of this is to:

 Determine hazards, their mitigations, and their traceability to requirements and to test protocols for a Computer System Development project

Hazards (1 of 3)

System Requirement ID	Hazard ID	User Exposure Risk	Regulatory Risk	Override Capability	Technical Risk	Security Risk	Consequence of Failure	Sum Value	Risk Assessment

- System Requirement ID Requirement ID from the CSRS
- **Hazard ID** a sequence number that uniquely identifies the Hazard (e.g., HA001, HA002)
- User Exposure Risk is the requirement used by a user?
 - If "Not Used," enter the value of one (1).
 - If "Infrequently," enter the value of one (1).
 - If "Occasionally," enter the value of four (4).
 - If "Frequently," enter the value of ten (10).
 - If "Insufficient Information," enter the value of ten (10).
- Regulatory Risk is data resulting from, or managed by, the function derived from the requirement is governed by GxP?
 - If "No," enter the value of zero (0).
 - If "Yes," enter the value of twelve (12).
 - If "Insufficient Information," enter the value of twelve (12).

@ 2010 Focus Compliance & Validation Services and SyberWorks, Inc.

Hazards (2 of 3)

System Requirement ID	Hazard ID	User Exposure Risk	Regulatory Risk	Override Capability	Technical Risk	Security Risk	Consequence of Failure	Sum Value	Risk Assessment

• **Override Capability** – could the function derived from the requirement be carried out by an alternative means?

- If "No," enter the value of zero (0).
- If "Yes," enter the value of ten (10).
- If "Insufficient Information," enter the value of ten (10).
- **Technical Risk** would the failure of the function derived from the requirement directly impact overall reliability of the system?
 - If "No Possibility," enter the value of one (1).
 - If "Low Possibility," enter the value of one (1).
 - If "Moderate Possibility," enter the value of four (4).
 - If "High Possibility," enter the value of ten (10).
 - If "Insufficient Information," enter the value of ten (10).
- **Security Risk** would the function derived from the requirement be controlled and secured through a user-ID and password and/or supervisory intervention?
 - If "No Possibility," enter the value of one (1).
 - If "Low Possibility," enter the value of one (1).
 - If "Moderate Possibility," enter the value of four (4).
 - If "High Possibility," enter the value of ten (10).
 - If "Insufficient Information," enter the value of ten (10).

Hazards (3 of 3)

System Requirement ID	Hazard ID	User Exposure Risk	Regulatory Risk	Override Capability	Technical Risk	Security Risk	Consequence of Failure	Sum Value	Risk Assessment

• **Consequence of Failure** – would the failure of the function derived from the requirement endanger the user, patient, or facility?

- If "No Possibility," enter the value of zero (0).
- If "Low Possibility," enter the value of one (1).
- If "Moderate Possibility," enter the value of eight (8).
- If "High Possibility," enter the value of twenty (20).
- If "Insufficient Information," enter the value of twenty (20).

• Sum Value –

 Add the values in columns, "User Exposure Risk," "Regulatory Risk," "Override Capability," "Technical Risk," "Security Risk," and "Consequence of Failure" columns, and enter the result in the "Sum Value" column

Risk Assessment -

- If the value is greater than or equal to 36, enter the value of "Major."
- If the value is greater than or equal to 15 but less than 36, enter the value of "Moderate."
- If the value is greater than or equal to zero but less than 15, enter the value of "Minor."

Mitigations

Hazard ID	Mitigation ID	Description

- Hazard ID Hazard ID designated in hazard table
- Enter a sequence number that uniquely identifies the Mitigation (e.g., MIT001, MIT002) in the "Mitigation ID" column. could be many to one relationship
- Description text that explains the "Major," "Moderate," or "Minor" assessment of the Hazard in the "Description" column:
 - If the Hazard is "Major," enter a brief narrative explaining how the Hazard will be mitigated.
 - If the Hazard is "Moderate," enter a brief narrative explaining how the Hazard will be mitigated.
 - If the Hazard is "Minor," enter "Hazard has been determined to be minor and will not be mitigated."

IQ/OQ/PQ

 "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..."

References

- General Principles of Software Validation; Final Guidance for Industry and FDA Staff, FDA, January 11, 2002
- GOOD MANUFACTURING PRACTICE GUIDE FOR ACTIVE PHARMACEUTICAL INGREDIENTS Q7, ICH, November 2000
- IEEE Standard for Software Test Documentation, September 16, 1998
- PHARMACEUTICAL QUALITY SYSTEM, Q10, ICH, June 4, 2008

Focus CVS Contact Information

9050 Executive Park Drive Suite A-202 Knoxville, TN 37923 USA E-mail: focus@focuscvs.com Office: (865) 694-7517 Fax: (865) 531-8854

www.focuscvs.com

@ 2010 Focus Compliance & Validation Services and SyberWorks, Inc.

Questions

2010 Focus Compliance & Validation Services and SyberWorks, Inc.