Standard Operating Procedure

Title: Computer Validation Master Plan

Department	Validation/Technical Services		Document no	VAL-210	
Prepared by:		Date:		Supersedes:	
Checked by:		Date:		Date Issued:	
Approved by:		Date:		Review Date:	

Table of Contents

Standard Operating Procedure

Title: Computer Validation Master Plan

1. Introduction

[***Enter company name***] manufactures and distributes a range of sterile and non-sterile, liquid, veterinary biological and pharmaceutical products from their sites [***address***].

The GMP facility has Code of Good Manufacturing Practice (cGMP) licenses with the **[*List all licenses*]**.

The core business focus is manufacturing and packaging of human and veterinary medicines in various forms. These include sterile injectable vaccines, oral, powders, creams, ointments, lotions, pastes and tablets and pour –on drenches.

This document aims to summarise the overall intentions and approach to the validation of automated system or computer system (known hereafter as computerised systems).

It is intended to be a working document and will be periodically updated by site management responsible for the execution of validation.

Systems addressed by this document are contained in the following list.
[***List of Computerised and Automated Systems***] used in the facility.

This Computer Validation Master Plan (CVMP):

- Identifies which computerised systems are subject to validation or qualification
- Identifies appropriate standards and guidelines to be referenced.
- Describes functional requirements for testing.

2. Purpose and Scope

The purpose of the Validation Plan will be to outline the principles and objectives of the Computer Validation Program for the [*site*]. Validation activities will be determined to be the sum of all activities that are conducted to ensure that the systems of manufacture are sufficient to produce products of a high quality that are safe for the intended user of the product.

The control of computer systems which impact GMP activities is important to assure control of processes, assessment of data, accuracy of manufacturing and control records and compliance with industry regulations.

This guideline provides guidance on how to validate computer systems that relate to GMP. This document should be considered as a guide; it is not intended to establish any mandatory or implied standard. Alternative approaches may be applicable.

This document will define the validation of all GxP computerised systems used by [company] including, where appropriate systems used by administration and quality groups (QA, QC, Regulatory Affairs Validation). This document does not define the requirements for the validation of Excel spreadsheets.

3. Regulatory Standards

[***Company***] agrees to comply with Good Validation Practices for computerised systems as defined in the code of GMP [***reference***].

4. Responsibility and Training

4.1 Validation Resources

[***Company***] will provide an appropriate level of competent resources to ensure the achievement of the outlined Validation program, with consideration of the risk to quality associated with the manufacture of products at [company] manufacturing sites.

4.2 Authority and Responsibility

The general authority and responsibilities for undertaken Validation projects are defined in the Validation Master Plan (VAL-080)).

Cat.	Description	Examples
1	**Infrastructure Software**	Operating systems, database managers, middleware, ladder logic interpreters, network monitoring etc.
2	**Previously (Firmware) no longer used**	N/A
3	**Non Configured Software Packages**	Packages are systems that are exposed to high volume use in the marketplace such as Microsoft office programs including spreadsheets or databases etc.
4	**Configured Software Packages**	Core components that can be configured by users. Each use of the standard product is specific for the particular user e. g. LIMS has database field labels, screens and reports configured by authorised users, ERP, DCS, SCADA, MES or Chromatography data systems.
5	**Custom Built Systems or 'Bespoke" systems**	Exclusively built solutions for a single or few customers e.g. PLC with single purpose dedicated program

Category 1 Infrastructure Software

Includes operating systems and layered software components such as database managers, middleware, and ladder logic interpreters. Also included are tools used to manage infrastructure, such as network performance monitors, batch scheduling tools, etc. This class is considered to be low risk due to two primary factors. First, infrastructure software is so ubiquitous that it is extremely unlikely that any unknown faults will exist. Second, this software is challenged indirectly in all other testing activities. While proper function of IT infrastructure may well be critical to satisfying a Critical Quality Attribute (CQA), infrastructure will almost always have an extremely low probability of failure. Applications built on top of this software may fail, but it will seldom be attributable to failure of infrastructure software.

Validation of infrastructure software is not required however records of the systems and their versions may be maintained in the MCSL. If a new version of the system software is required, a review should be conducted to determine the possible impact of the changes on the existing software application(s), and system configuration files.

Category 2 (Firmware) is no longer a separate category since modern firmware can be so sophisticated that there is no longer any justification for differentiation. Firmware can fit into any of the categories depending on the nature of the embedded software.

Category 3 Non-Configured Software (NCS) or Standard Software

This category has been expanded to include many examples of firmware. Non-Configured in this sense refers to configuration to meet the needs of a business process; run-time parameters can still be configured. Off-the-shelf software has grown in sophistication to the point where some examples are configurable to meet the business process, and hence could be considered Category 4.

Equipment qualification or verification documentation will require a reference to firmware versions and configuration settings. A validation assessment is required for all firmware upgrades. The qualification of operation and control functions of firmware should be considered as part of new equipment qualification.

User-derived functionality developed using non-configured software (e.g. a report, macro or a program) require validation if it involves GMP aspects. Implementing new versions of a NCS package require assessment of the user-derived functionality with respect to impact on data integrity, security, algorithms, high-level language code and macros. Additionally, due to the ease by which changes can be made by the user, formal change control is imperative.

Categories 4 - Configured Software

These software packages are called configured software and include Distributed Control Systems (DCS), Supervisory Control and Data Acquisition packages (SCADA), manufacturing execution systems and some LIMS and MRP packages. In these examples the system and platform should be well known and mature

6. Software Development and Validation – Category 4 and 5 Software

A User Requirements Specification (URS) shall be prepared specifying the objectives of any proposed GxP related computer system, the data to be entered and stored, the flow of data, the information to be produced, the limits of any variables and the operating program(s) and test programs, together with examples of each document produced by the program, instructions for testing, operating and maintaining the system and the names of the person or persons responsible for its development and operation.

The development, implementation and operation of a category 4 or 5 GxP related computer system shall be carefully documented at all stages and each step proven to achieve its written objective under challenging test conditions.

Software developers of Category 4 and 5 systems should follow a Software Quality Assurance Plan (SQAP) and should be audited if possible for adherence to the plan. Vendors should be asked to provide written assurance that software development or modification has followed the SQAP or an equivalent system.

A logic flow diagram or schematic for software should be prepared for critical evaluation against system design / requirements / criteria.

Records should be available for the following aspects of a computer system validation:

- Protocol for validation
- General description of the system, the components and the operating characteristics
- Diagrams of hardware layout/interaction
- List of programs with brief description of each
- System logic diagrams or other schematic form for software packages
- Current configuration for hardware and software
- Review of historical logs of hardware and software for development, start-up and normal run periods
- Records of evaluation data to demonstrate system performs as intended (verification stage and ongoing monitoring)
- Backup and recovery procedures
- Range of limits for operating variables
- Details of formal change control procedure
- Records of operator training
- Details of access security levels/controls
- Procedure for ongoing evaluation

Individual validation protocols for all qualifications and validations will define the actual validation testing which will be required.

7. System Documentation

A written detailed description of the system should be produced (including diagrams as appropriate) and kept up to date. It should describe the principles, objectives, security measures and scope of the system and the main features of the way in which the computer is used and how it interacts with other systems and procedures.

8. Security of Computer Systems

A hierarchy of permitted access to enter, amend, read, or print out data shall be established and documented according to user need.

Suitable methods of preventing unauthorised entry shall be available, such as pass cards or personal user-identity codes.

The recovery procedure to be followed in the event of a system breakdown shall be defined in writing. This procedure should be designed to return the system to a previous state.

Methods of recovery shall be validated based on the criticality of the system.

