

**COMPUTER SYSTEMS
VALIDATION
IN
CLINICAL RESEARCH**

A Practical Guide

ACDM/PSI Working Party

The Association for Clinical Data Management (ACDM) and Statisticians in the Pharmaceutical Industry (PSI) were approached by the Association of the British Pharmaceutical Industry (ABPI) to investigate the need for an industry-wide guideline on computer systems validation in clinical research. A joint Working Party was formed in August 1995 with the remit to draw up a guideline. The authors of the Guideline are the members of this Working Party and comprise five ACDM members, two PSI members and two members with a Regulatory/Quality Assurance background as listed below:

- Andrew Gold** (PA Consulting Group)
- Nicky Jakeman** (Pfizer)
- Sue Mayes** (AstraZeneca)
- Fiona Portwood** (SPS)
- Alan Rawling** (Assured Systems)
- John Shelton** (Wyeth)
- Jane Tucker** (Wyeth)
- Heather Wells** (Stacion International)
- Louise Wood** (Searle).

Acknowledgements

The ACDM/PSI Working Party wish to acknowledge the interest and contribution from many leading figures and interested parties in the area of clinical computer systems validation, within the UK, Europe and North America. Many constructive comments were received as a result of the Expert Review meeting held in April 1997 and the joint ACDM/PSI open meeting held in November 1997. The Working Party thank all contributors for their input and support.

CONTENTS

- EXECUTIVE SUMMARY 7**
- 1. KEY DEFINITIONS 9**
 - 1.1 **Clinical Research Computer Systems 9**
 - 1.2 **Validation 9**
- 2. INTRODUCTION 11**
 - 2.1 **Regulatory Background 11**
 - 2.2 **Scope and Benefits of Validation 11**
 - 2.3 **Objectives of the Guideline 12**
 - 2.4 **Scope of the Guideline 12**
 - 2.5 **Structure and Use of the Guideline 12**
- 3. VALIDATION POLICY 17**
 - 3.1 **Objectives and Scope of a Validation Policy 17**
 - 3.2 **Responsibilities 17**
 - 3.3 **Personnel 18**
 - 3.4 **Documentation. 18**
 - 3.5 **Management of Validation 18**
 - 3.6 **Decommissioning 18**
 - 3.7 **Quality Management 18**
- 4. VALIDATION SOPS 19**
- 5. REACHING THE VALIDATED STATE. 21**
 - 5.1 **Validation Process 21**
 - 5.1.1 **Systems Developed In-House. 23**
 - 5.1.2 **Systems Developed by Third Parties (Vendors). 23**
 - 5.1.2.1 **Elements Under Vendor Control. 23**
 - 5.1.2.2 **Elements Under User Control 24**
 - 5.1.2.3 **Escrow Agreements 24**
 - 5.1.3 **Commissioned Systems 24**
 - 5.2 **Legacy Systems 25**
- 6. MAINTAINING THE VALIDATED STATE. 27**
 - 6.1 **Security 27**
 - 6.2 **Operational Management. 28**
 - 6.3 **Business Continuity 28**
 - 6.4 **Change Management 28**
 - 6.5 **Periodic Review 31**
 - 6.6 **Decommissioning 31**
- 7. ONE-OFF AND STANDARD PROGRAMS 33**
 - 7.1 **Responsibilities 33**
 - 7.2 **Documentation. 33**
 - 7.3 **Software Standards 33**
 - 7.4 **Coding Review 34**
 - 7.5 **Testing 34**
 - 7.6 **Change Control 34**



8.	QUALITY MANAGEMENT	35
8.1	Quality Control (QC)	35
8.2	Quality Assurance (QA)	35
8.2.1	In-House Audits	36
8.2.2	External Audits	36
8.2.2.1	CRO Audits (including Laboratories)	36
8.2.2.2	Vendor Audits	37
8.2.3	Investigational Site Audits	37
9.	SPECIFIC CLINICAL SYSTEMS	39
9.1	Randomisation Systems	39
9.2	Data Capture Systems	39
9.2.1	Automatic Measuring Devices	39
9.2.2	Manual Data Input	40
9.2.2.1	In-House Data Entry Systems	40
9.2.2.2	Remote Data Entry Systems	40
9.2.2.3	Electronic Diary Cards	41
9.2.3	Automated Data Input	42
9.2.3.1	Optical Character Recognition (OCR) and Intelligent Character Recognition (ICR).	42
9.2.3.2	Bar Coding Systems	42
9.3	Electronic Transfer of Data and/or Software	42
9.4	Clinical Database Management Systems (CDMS)	43
9.4.1	Database Set-up	43
9.4.2	Data Capture	44
9.4.3	Derived Data	44
9.4.4	Data Checking	44
9.4.5	Audit Trail	45
9.4.6	Database Snapshot	45
9.4.7	Data Review	45
9.4.8	Reporting	45
9.4.9	Database Locking/Securing	45
9.4.10	Archiving and Retrieval	45
9.5	Derived Data	46
9.6	Drug Supplies Accountability Systems	46
9.7	Statistical Systems	47
9.7.1	Commercial Systems	47
9.7.2	In-House and Commissioned Systems	47
9.8	Computer Aided Review Tools	48
9.9	Electronic Document Management Systems (EDMS)	49
9.9.1	Optical Imaging	49
9.9.2	Electronic Document Repository	50
9.9.2.1	Electronic SOPs	50
9.9.3	Indexing	50
9.9.4	Work Flow Systems	51
9.9.5	Archiving	51
9.9.6	Document Sign-off/Authorisation	51
9.10	Electronic Records and Electronic Signatures	52
9.11	Electronic Publishing and Regulatory Submissions	53
9.12	Pharmacovigilance Systems	53
9.13	Administrative Systems	53
10.	GLOSSARY	55
11.	REFERENCES	58
12.	BIBLIOGRAPHY	59
13.	SOME USEFUL WEB SITES	60

FIGURES

- 1. Validation Process Overview16
- 2. Reaching the Validated State Flow Chart22
- 3. Change Management Flow Chart30

TABLES

- 1. Sections of particular interest for different types of system14
- 2. Sections of particular interest for specific personnel15

APPENDICES

- 1. Regulatory Guideline References to Validation61
- 2. Some Common Development Life Cycle (SDLC) Models62
- 3. User Acceptance Testing66
- 4. SOP Contents67
- 5. Documentation Checklist70
- 6. Potential Causes of Validation Failure73
- 7. Validation Plan Contents74
- 8. Sample Programming Standards75
- 9. Guidelines for Testing One-Off Programs78
- 10. Published Standards and Guidelines for Systems/Software Development and Validation80
- INDEX82



EXECUTIVE SUMMARY

EXECUTIVE SUMMARY

Validation encompasses the entire system development life cycle from initiation through development, testing and production use to decommissioning. It is a process which demonstrates that a system is developed, used, maintained, evolves and is eventually decommissioned in a controlled, documented manner. A broad view of computer systems is taken to cover software, hardware, processes and people. The purpose of the Guideline is to suggest approaches by which validation may be achieved.

Validation of clinical research computer systems is required by ICH regulatory guidelines on Good Clinical Practice and Statistical Principles for Clinical Trials. However the over-riding rationale for validation is that it makes good business sense by ensuring quality, timeliness and efficiency, and by addressing business risks.

This guideline is structured as an aid to both the novice and the experienced practitioner. It describes how to embrace validation within the culture of an organisation via a high-level policy, SOPs and clearly identified accountability. While corporate management has ultimate responsibility for systems validation, specific responsibility for the implementation of the validation process rests with the user management. The generalities of reaching and maintaining the validated state are covered for any system from any source, from a one-off analysis program to a large multi-functional, multi-site installation, and from inception to decommissioning.

Aspects of different types of clinical research computer systems which require special attention during validation are highlighted. Such systems include administration, randomisation, drug supplies, data capture and transfer, databases and associated review tools, statistical analysis, document management, publishing and regulatory submission, and pharmacovigilance.

Throughout the guideline emphasis is placed on the importance of procedures, training, adequate documentation and approval signatures to provide evidence that a system has been, and remains, properly validated.



KEY DEFINITIONS

I. KEY DEFINITIONS

I.1 Clinical Research Computer Systems

For the purposes of this guideline, a clinical research computer system is defined as:

The set of hardware, software, procedures and people which together perform one or more of the capture, processing, analysis and reporting functions on clinical trial data and management of the clinical development programme.

I.2 Validation

Among the many published definitions of validation, the following two are useful and have regulatory sources.

- 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. [1]
- The demonstration that a computerised system is suitable for its intended purpose. [2]

For the purposes of this guideline, the above definitions have been combined and extended as follows:

The process whereby documentary evidence is established which:

- (a) *demonstrates that a system was developed and implemented, and is operated and maintained, in a controlled manner throughout its life-time up to and including decommissioning and*
- (b) *results in a high degree of assurance that the system consistently meets its specification, and is therefore suitable for its intended purpose.*



INTRODUCTION

2 INTRODUCTION

The process of system validation has the ultimate goal of demonstrating that a quality system has been produced and that changes to the system are well controlled. The computer system validation process is a fundamental part of the overall business process. It will ensure the integrity of clinical trial data provided the overall business is driven by sound quality management processes. For example, the requirement for the commitment of senior management to system validation within the context of Good Clinical Practice within the sponsor company should extend to suppliers and investigator sites. No matter how rigorously the validation process is implemented in-house, the quality of the final product may be compromised unless there are adequate controls of external processes.

Individual circumstances may influence the interpretation and application of the recommendations in this guideline and the extent to which the recommended approach is adopted is a matter of risk assessment on the part of each company. The guiding principle should be the ability to demonstrate that computer systems remain under control. Factors such as resource, type of system, regulatory impact and the current quality procedures of the company will influence an individual company's approach.

To aid use of the Guideline see Section 2.5 and Figure 1.

2.1 Regulatory Background

The need for validation of computer systems in clinical research is documented in the ICH regulatory guidelines for Good Clinical Practice (GCP) (E6) [3] and Statistical Principles for Clinical Trials (E9) [4]. The FDA draft proposal 'Guidance for Industry: Computerized Systems Used in Clinical Trials' establishes in further detail specific requirements [5].

2.2 Scope and Benefits of Validation

Validation encompasses the entire system development life cycle from initiation through development, testing and production use to decommissioning.

In addition to the need for regulatory compliance, validation makes good business sense and demonstrates that a system is under control. By assuring the integrity of clinical trial data from paper or electronic source to regulatory submission and beyond, validated systems improve the quality and speed of submissions and should lead to earlier regulatory approval. Credibility is enhanced and the need for rework due to system errors is reduced.

2.3 Objectives of the Guideline

This document is intended to serve both as a reference for the experienced user and as a helpful guideline for the novice. It covers both the general principles of validation and guidance on their application to specific clinical research applications. The guideline is not intended to be prescriptive; there will always be the need to interpret and adapt it to the user's own environment in order to address specific business risks and regulatory requirements.

2.4 Scope of the Guideline

The guideline addresses all computerised clinical systems used directly or indirectly by a sponsor to capture, process, analyse, and report clinical data, and to manage clinical development programmes, before and after regulatory submission. Although validation of the latter systems is not required by regulatory guidelines, it makes good business sense. Systems covered range from single programs for specific studies to multi-module international applications and include both in-house and commercial systems. Indirect use encompasses systems used by Contract Research Organisations (CROs) and investigator-owned systems. It is generally considered that there is no requirement for validation of commercial hardware and established operating systems or for packages such as the SAS system, Oracle and MS Excel, as entities in their own right. However, most are configurable systems and so need adequate control of installation and their configuration parameters. Validation of user applications based on these products should ensure that the underlying system functions as intended within the conditions of its use by the application. Validation should also recognise the overall IT infrastructure in which the system operates.

2.5 Structure and Use of the Guideline

Sections 1-12 comprise the core document. Sections 1-8 provide generic principles, which are interpreted for specific clinical research applications in Section 9. A Glossary is included as Section 10, and References and Bibliography as Sections 11 and 12 respectively. Section 13 directs the reader to some useful web sites.

The ten appendices provide extracts from regulatory guidelines, additional specific guidance, check lists and a list of published standards on system development and validation.

The guideline is intended to provide guidance for all those with an interest in the validation of computer systems in clinical research. These will include everyone from the single user with a small stand-alone system or one-off program to validate, to board members who need guidance on the requirements for a corporate validation policy and the validation requirements for their systems.

For small companies where personnel involved in the validation of a system might perform different roles, the divisions between developer, user and system manager may become blurred. There will always be a need to interpret guidelines such as these in a pragmatic way while assessing the risk to the validation of the system should, for example, the developer also perform the testing.

Table 1 - lists sections which are likely to be of particular interest to those involved in the validation of certain types of system.

Table 2 - lists sections which are likely to be of particular interest to certain groups of people with responsibility for computer system validation in clinical research.

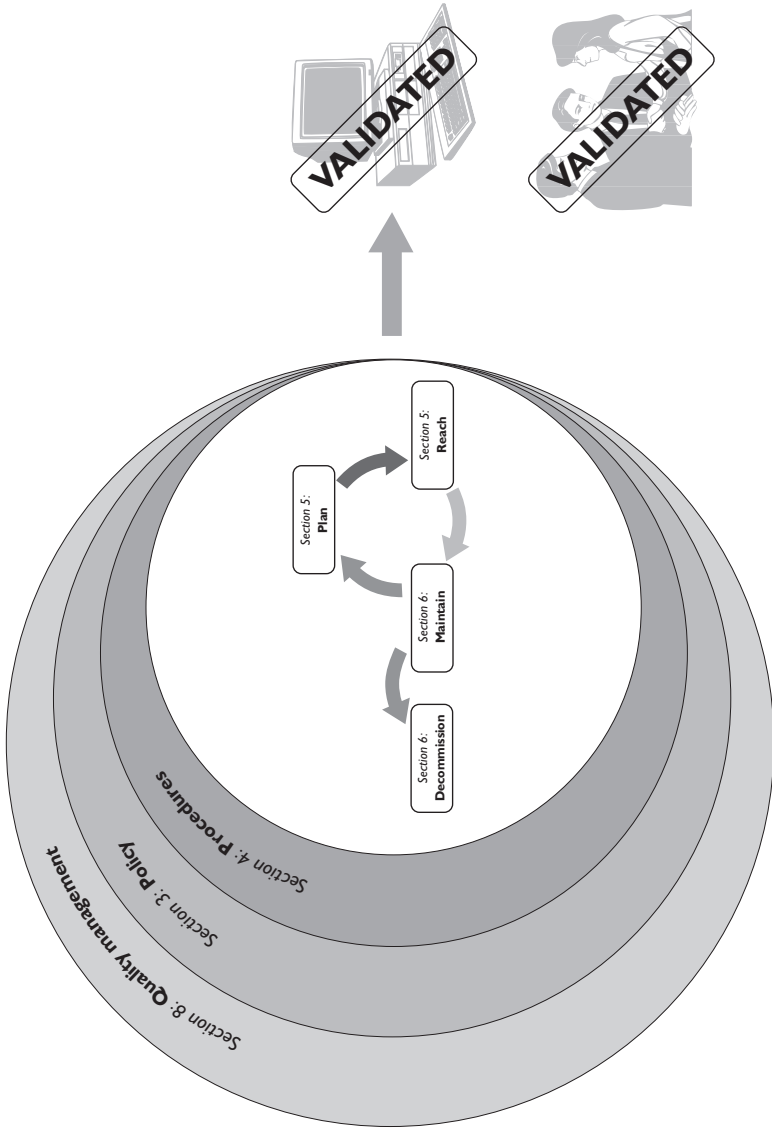
Table 1 Sections of particular interest for different types of system

Section	Section Title	Third party System	In-house developed system	Legacy system	One-off programs
1	Key Definitions				
2	Introduction				
3	Validation Policy				
4	Validation SOPs				
5	Reaching the Validated State	✓	✓	✓	
6	Maintaining the Validated State	✓	✓	✓	✓
7	One-Off and Standard Programs				✓
8	Quality Management	✓			
9	Specific Clinical Systems	✓	✓	✓	✓
10	Glossary				
11	References				
12	Bibliography				
13	Some Useful Web Sites				
Appendix 1	Regulatory Guideline References to Validation				
Appendix 2	Some Common SDLC Models		✓		
Appendix 3	User Acceptance Testing	✓	✓	✓	
Appendix 4	SOP Contents				
Appendix 5	Documentation Checklist	✓	✓	✓	
Appendix 6	Potential Causes of Validation Failure		✓		✓
Appendix 7	Validation Plan Contents	✓	✓	✓	
Appendix 8	Sample Programming Standards				✓
Appendix 9	Guidelines for Testing One-Off Programs				✓
Appendix 10	Published Standards				

Table 2 Sections of particular interest for specific personnel

Section	Section Title	Software developers and IT professionals	System User management	QA	End-user programmers
1	Key Definitions	✓	✓	✓	✓
2	Introduction	✓	✓	✓	✓
3	Validation Policy	✓	✓	✓	✓
4	Validation SOPs		✓	✓	✓
5	Reaching the Validated State	✓	✓		
6	Maintaining the Validated State	✓	✓		
7	One-Off and Standard Programs				✓
8	Quality Management		✓	✓	
9	Specific Clinical Systems		✓		
10	Glossary				
11	References				
12	Bibliography				
13	Some Useful Web Sites				
Appendix 1	Regulatory Guideline References to Validation		✓	✓	✓
Appendix 2	Some Common SDLC Models	✓			
Appendix 3	User Acceptance Testing		✓		
Appendix 4	SOP Contents	✓	✓	✓	✓
Appendix 5	Documentation Checklist	✓	✓	✓	
Appendix 6	Potential Causes of Validation Failure	✓	✓	✓	✓
Appendix 7	Validation Plan Contents	✓	✓		
Appendix 8	Sample Programming Standards				✓
Appendix 9	Guidelines for Testing One-Off Programs				✓
Appendix 10	Published Standards	✓	✓		

Figure 1 Validation Process Overview



VALIDATION POLICY

3. VALIDATION POLICY

A corporate validation policy document should define and standardise the approach to validation. The policy should give reasons why validation is important and embrace the following fundamental principles:

- An identified user, not supplier, is responsible for validation (*see Section 3.2*)
- Validation should be reasonable, practical and add value to the business.
- Validation of new systems should always be prospective and have signed and dated approval.
- Development of a computer system should follow a specified methodology, e.g. classical software development life cycle (SDLC), rapid application development (RAD) or prototyping (*see Appendix 2*).
- Validation does not end with the introduction of a new system but continues through operational use to decommissioning.

The following sections provide a checklist of the major elements of a comprehensive validation policy together with guidance on content.

3.1 Objectives and Scope of a Validation Policy

The objectives and scope of the policy should be defined. They should state:

- the aims of validation
- the need for minimisation of business risk
- the extent to which different categories of system will be validated
- the principles by which validation will be achieved and maintained.

3.2 Responsibilities

While corporate management has ultimate responsibility, the policy should clearly identify specific responsibilities for ensuring that the validation policy is implemented consistently, including:

- designation of suitably qualified and trained personnel to undertake the validation process.
- establishment of appropriate procedures
- assurance that systems are validated appropriately.

Responsibility for demonstrating that a specific system has been validated should be assigned by corporate management to the system's user management. However, all personnel involved in the development and operation of the system should be aware of their validation responsibilities. Mechanisms should be identified to ensure that the

policy and all related standard operating procedures (SOPs) are distributed to all staff responsible for initiating or purchasing new systems, or for making changes to systems, their operating environments or their associated tools and software, and that these staff are trained in their implementation and remain compliant. Where an individual undertakes multiple roles, these should be clearly defined and the associated risks to validation addressed.

3.3 Personnel

In line with the principles contained in GCP guidelines, the policy should state that all personnel involved in any aspect of validation should have:

- a curriculum vitae (CV)
- sufficient education, training and experience to enable them to undertake their assigned functions
- a documented record of training, experience and competency which is regularly updated
- an up-to-date job description.

3.4 Documentation

The policy should emphasise the need for documentation to be version controlled including status (e.g. draft, under revision, approved), complete, clear, accurate, intelligible to the intended audience and approved, be it technical or non-technical (*see Appendix 4: SOP on documentation management*). A documentation checklist is given in Appendix 5.

3.5 Management of Validation

The policy should state how validation is managed. A project management approach is recommended. The policy should address:

- Systems Developed In-House (*see Section 5.1.1*)
- Systems Developed by Third Parties (*see Section 5.1.2*)
- Commissioned Systems (*see Section 5.1.3*)
- Legacy Systems (*see Section 5.2*)
- One-off and Standard Programs (*see Section 7*)
- Working with CROs (*see Section 8.2.2.1*)

The policy should indicate how the validated state will be maintained during production use and in the event of changes to the system (*see Section 6*).

3.6 Decommissioning

The policy should define the approach to be taken to decommissioning systems, taking into account requirements for subsequent access to archived data (*see Sections 6.6 and 9.9.5*).

3.7 Quality Management

The policy should describe the role of, and identify responsibilities for, quality control (QC) and quality assurance (QA) in validation activities to provide management with assurance that quality is built into the system in compliance with SOPs (*see Section 8*).

VALIDATION SOPs

4. VALIDATION SOPs

For the purposes of the Guideline, the term SOP (standard operating procedure) refers to any documentation of a process. For effective development, testing, implementation, operation, maintenance and decommissioning of any computer system, there should be a number of SOPs in place, compliant with regulations and policies, covering the following areas:

- Documentation management
- Software development and testing including one-off programs *(see Section 7)*
- System set-up and installation
- User acceptance testing *(see Appendix 3)*
- Training
- Security *(see Section 6.1)*
- System use and maintenance
- User support
- Problem management
- System back-up and restoration *(see Section 6.1)*
- Business continuity *(see Section 6.3)*
- Change management *(see Section 6.4)*
- Periodic review *(see Section 6.5)*
- Decommissioning *(see Section 6.6)*
- Archiving and retrieval *(see Section 6.2 and 9.9.5)*
- Audit and review *(see Section 8.2)*

The first time validation is performed it may not be in accordance with authorised SOPs but any validation activity should be planned and this plan should be documented. This plan may form the basis for the SOP.

Suggested SOP contents under each of the above headings are given in Appendix 4. Their number, titles and format will depend on each company's structure, policies and philosophies. However, all SOPs should clearly identify the scope, specific tasks and responsibilities, and the flow of the tasks within the process. They should also be subject to an SOP on format, content, version control, approval and distribution. For SOPs to be effective, appropriate training must be provided in each SOP and documented. All staff should have the ability to request changes to ensure that the SOPs which are written are workable and will be adhered to.

There should be a procedure in place to ensure that any changes to regulations or policies are incorporated into SOPs within a specified time frame.



REACHING THE VALIDATED STATE

5. REACHING THE VALIDATED STATE

This section suggests how to approach the initial validation of large multi-functional systems and retrospective evaluation of existing systems ('legacy systems'), developed in-house or purchased from third parties (including commissioned systems). The process of validation aims to anticipate the ways in which a system could fail to produce the expected result, and to put measures in place to reduce this risk. A significant obstacle to successful validation is the failure to recognise the requirement for it early enough, if at all. A list of the potential causes of validation failure is given in Appendix 6. Once achieved, the validated status of a system will need to be maintained. Guidance on maintenance is provided in Section 6.

Much of the validation process consists of normal project management activities, considered from a validation viewpoint, and can be readily incorporated into existing working practice. In some sectors of industry, the term 'validation' is used synonymously with 'acceptance testing', and it is worth emphasising the wider meaning adopted in this guideline, which encompasses all aspects of system development and implementation (*see Section 5.1*). Sound documentation, essential to demonstrate validity, is the key to successful validation. Useful maxims in this context are *'If it isn't documented, it's a rumour'* and *'Say what you do, do as you say and have evidence to prove it'*.

The principles described can be applied equally to traditional software development life cycles, rapid application development or prototyping (*see Appendix 2*). For the latter, some documentation (e.g. software specifications, testing, system environment) may be produced later in the development cycle, after the system specification has stabilised. However, provided that the system is validated in accordance with the validation plan (*see Section 5.1.1 and Appendix 7*) prior to production delivery, there should be no special problems.

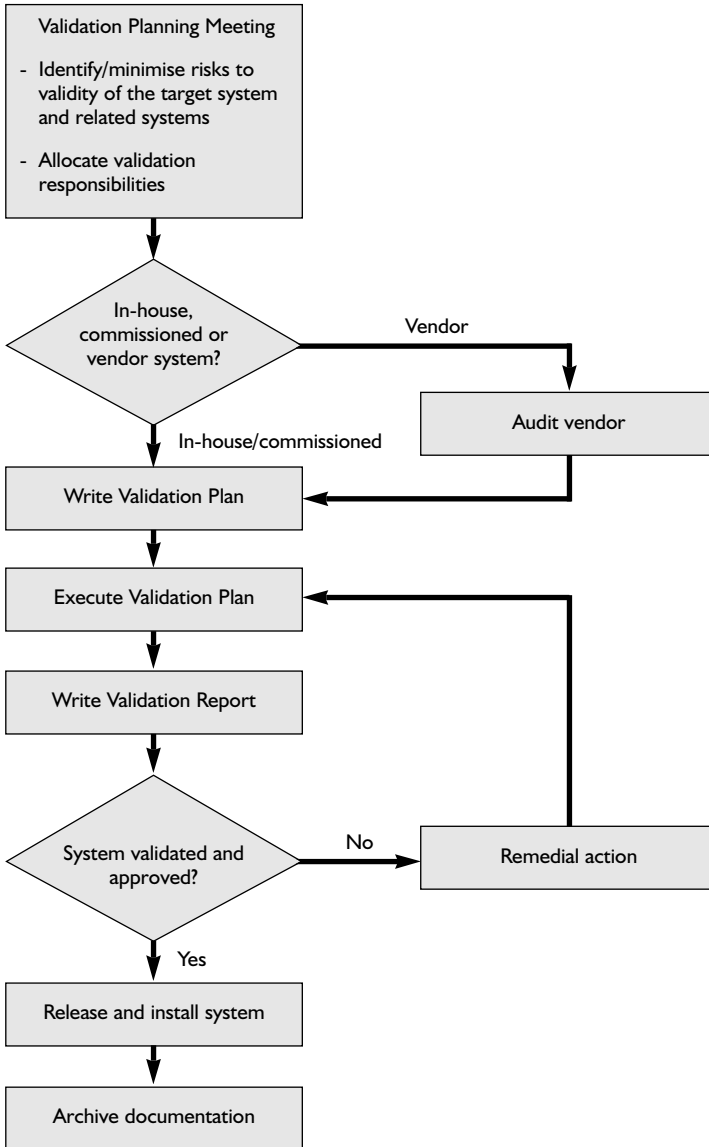
To avoid the risk of an unvalidated system being used in production, or of a system being changed during testing, there should be separate development, user acceptance testing and production computing environments. A system, or changes to a system, should not be released for production use until they have been validated and approved.

5.1 Validation Process

The validation process will depend on whether the system to be validated is to be developed internally, supplied by an external vendor, or is a commissioned system. The SOP for the validation process should cover all cases.

Each process is described in one of the sections below. See also Figure 2.

Figure 2 Reaching the Validated State Flow Chart



5.1.1 Systems Developed In-House

For in-house systems, validation should be considered at the very outset of a new project. The following steps are suggested (*see Figure 2*):

1. At a validation planning meeting of all those involved in the project (e.g. IT and development personnel, QA, users):
 - (a) List all identifiable risks to system validity, e.g. inadequacies in SOPs, system specification, change control, testing, training, etc.
 - (b) Determine the measures to be taken to reduce each risk, e.g. system specification, testing, training strategy, etc.
 - (c) Define the documents needed to demonstrate that the measures have been taken (*see Appendix 5*)
 - (d) Allocate responsibilities for writing, review and approval of the documents.
2. After the meeting, finalise and approve a validation plan describing the measures and responsibilities agreed, any assumptions about or limitations on the extent of validation, and justification for any exclusions. Suggested validation plan contents are given in Appendix 7.
3. Circulate the validation plan to all those having responsibility for any aspect of validation.
4. As the project proceeds, collect together final approved versions of the documentation specified in the validation plan.
5. At the end of the project, write a validation report either to:
 - (a) confirm that all the agreed documentation has been received in its final form,
 - (b) recommend remedial action, or
 - (c) justify, as an acceptable risk, acceptance of the system with acknowledged defects.
6. Install and release.
7. Archive the validation document set, such that it can be readily retrieved for audit when required.

5.1.2 Systems Developed by Third Parties (Vendors)

It is the responsibility of the intending user to establish the validity of software purchased off-the-shelf from a third party. Certain elements of a third party system will be under the control of the vendor, while others will be under the control of the user. Validation of each is considered separately below.

5.1.2.1 Elements Under Vendor Control

Validation of elements under vendor control should take the form of an audit (*see Section 8.2.2.2*). While the objective of validation is unchanged, the validation document set will not have been agreed in advance, and information may have to be drawn from disparate documents. Elements under the control of the vendor, for which documented evidence should be sought, include:

- management of software development personnel
- formal software development life cycle and associated documentation
- programming standards (*see Appendix 8*)
- software fault management
- documentation management
- configuration control
- user manuals
- release notes
- user support
- upgrade provision mechanisms.

If the resulting documentation set is insufficient to validate the system, and the deficiencies cannot be remedied, then the system should not be used.

5.1.2.2 Elements Under User Control

Elements under the control of the user should be validated prospectively, following the procedure described in Section 5.1.1. These elements include:

- system installation
- operating environment
- user acceptance testing (*see Appendix 3*)
- user training
- business continuity (*see Section 6.3*)
- change control procedures for the production system (*see Section 6.4*)
- reporting of software faults.

5.1.2.3 Escrow Agreements

Another important point to consider with vendor-supplied software is sponsor access to proprietary source code, which can be achieved by an escrow agreement. This is a legal agreement, whereby the source code is lodged by the vendor with a third party. The source code and associated documentation are only made available to the sponsor when a certain condition is met, e.g. the vendor goes out of business.

5.1.3 Commissioned Systems

Where a system is to be supplied by a third party to the specification of the purchaser, it should be possible to carry out a completely prospective validation, as for in-house systems, providing the requirement for validation is identified at the outset. As with all systems, the responsibility for validation lies with the intending user.

Agreement should be reached with the vendor on the documentation set necessary to establish validity of those parts of the system under vendor control. In addition, the vendor should agree to inspection either by the user, or if this would compromise the vendor's commercial interests, by an independent validation expert, employed by the user.

5.2 Legacy Systems

Legacy systems (i.e. systems in use but with inadequate or no evidence of validation) should be assessed by collecting relevant documentation and identifying validation deficiencies (retrospective evaluation). If system documentation is not available, a description of the system should be produced. A plan should then be established for any additional testing and documentation. The next upgrade or major change to a system is a good opportunity to address any deficiencies. An evaluation plan should contain all the appropriate sections of a validation plan in a similar format (*see Section 5.1.1 and Appendix 7*). This plan should be reviewed by an appropriate authority, agreed and signed off as complete. If the system is in regular use, one complete cycle of use of the system should be identified for evaluation purposes. During this cycle the documentation in the evaluation plan should be produced at each stage, checked and, if correct, signed off.

Once the evaluation of each stage is successfully completed, an evaluation report can be produced containing all the documentation and a statement of successful completion. This report and all supporting documentation should be archived. If any evaluation stage fails, the failure should be documented with details of the corrective action taken. Once this action has been taken the system can then be re-evaluated in the same way as any other system after upgrade/correction (*see Section 6.4*).

Once a legacy system has achieved a satisfactory, documented evaluated state, any subsequent changes can be validated in the same way as any other system (*see Section 6.4*).



MAINTAINING THE VALIDATED STATE

6. MAINTAINING THE VALIDATED STATE

The validated status of any system is subject to threat from changes in its operating environment, either known or unknown. Procedures relating to security, operational management, business continuity, change management, periodic review and decommissioning should be in place to minimise such risks. Some of these procedures may not be under the direct control of the users. For those systems either developed by a third party or commissioned, some aspects of the system may be managed by the vendor, and these should be documented within a Service Level Agreement. However, user management should ensure that the procedures are adequate for the systems concerned.

6.1 Security

Hardware, software and data (local or remote) should be protected against loss, corruption and unauthorised access. Physical security is the prevention of unauthorised physical access by internal or external personnel to computer system hardware. Logical security is the prevention of unauthorised access to software applications and data. Access control should be provided by the network and/or application software. Procedures should be in place to ensure that:

- Rules and responsibilities for assigning access rights are defined
- Rules for password definition exist
- Passwords are subject to regular change
- Use of electronic signatures is adequately controlled (*see Section 9.10*)
- Default access for any user to perform any function is NO access
- Access is granted at the lowest possible level
- Access rights are documented and reviewed regularly to ensure they are appropriate
- All users receive appropriate training in the use of the system
- The system is protected from viruses
- Electronic links used to transfer data are secure.

Routine back-up of software and data should be performed to enable restoration of recent copies in case of a loss of work (*see Section 6.3*). A register of back-up activity should be kept. Back-up copies should be stored in off-site or remote facilities subject to stringent security and environmental controls.

6.2 Operational Management

Routine use of the system requires the following procedures to be in place in order for it to be operated as intended:

- Training: all staff involved in the system whether developers, ‘system managers’ or users (e.g. clinical data managers, statisticians) should receive training in and have documented competency in their areas of expertise.
- System Use and Maintenance.
- User Support and Problem Management: enabling reporting and registration of any problems encountered by users of the system. These can be filtered according to whether their cause lies with the user or the system, and fed back appropriately. Those problems which require a possible system change are then managed through a ‘Change Management’ procedure (see Section 6.4).
- Archiving: all validation documentation should be archived in a facility which is both secure and where possible protected from environmental hazards and maintained at appropriate humidity levels, etc.. Archiving electronic copies of software and data at such time points may also be appropriate. A record of all archived material should be maintained.
- Quality Control to pick up possible system errors as well as human error or misuse (see Section 8.1).

6.3 Business Continuity

Business continuity procedures, including disaster recovery, should ensure minimal disruption in the event of loss of data or any part of the system. It is necessary to ensure that the integrity of the clinical data is not compromised during the return to normal function. At its lowest level, this may mean the accidental deletion of a single file, in which case a procedure should be in place to restore the most recent backed-up copy. At the other extreme a disaster such as a fire could result in loss of the entire system. For this situation a procedure addressing the following should be in place:

- specification of the minimum replacement hardware and software requirements and their source
- specification of the time frame within which the replacement system should be in production, based on business considerations
- implementation of the replacement system
- steps to revalidate the system to the required standard
- steps to restore the data so that processing activities may be resumed as soon as possible.

The procedure employed should be tested regularly and all relevant personnel should be made aware of its existence. A copy of the procedure should be maintained off-site.

6.4 Change Management

It is necessary to control changes to a system in order to ensure that it continues to function correctly. Procedures should be in place to identify, authorise, validate and install changes to the system. These include changes made to:

- hardware



- software, e.g. a specific application
- the environment in which it operates, e.g. the operating system
- the configuration of the hardware/software
- the use of the system.

Changes to a system may be proposed either to correct an error or to enhance functionality or performance. The following points should be considered:

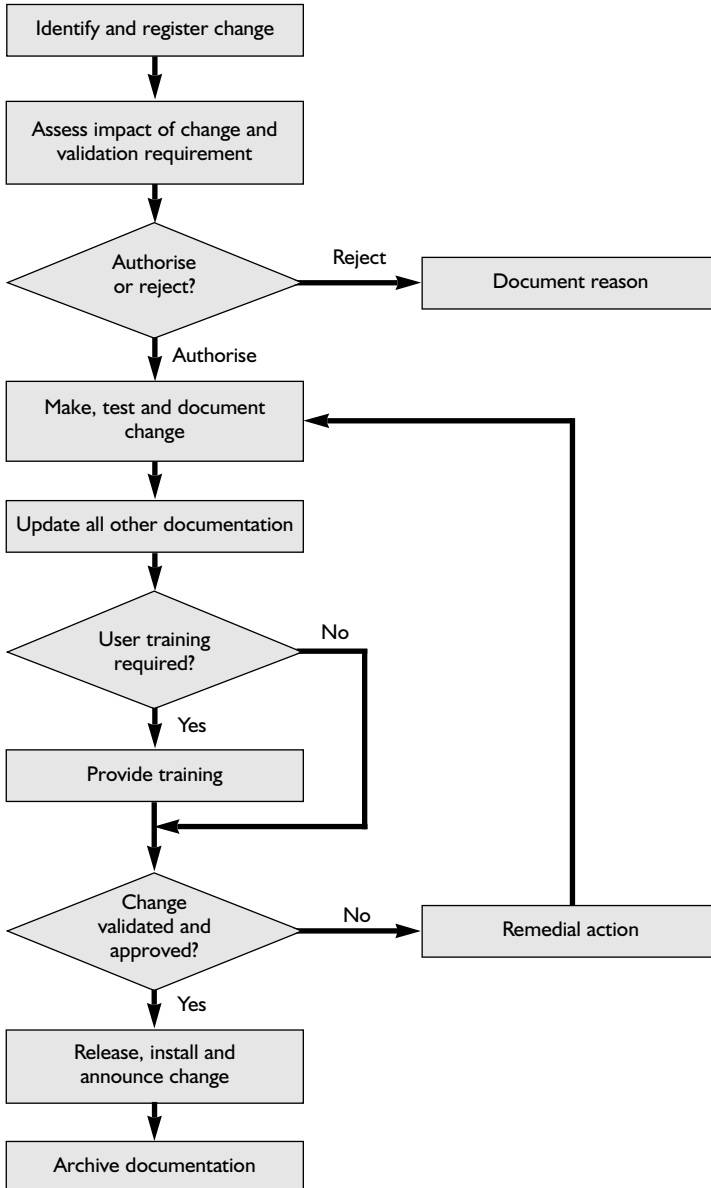
- Proposed changes should be documented in a register from which the status of all current and historical proposals can be ascertained.
- Impact assessment of the effect of a change on other parts of a system should be performed.
- Approval for a change should be formally documented and signed-off by the system's user management.
- If authorisation is not given, the reason should be documented.
- Authorised changes to software should be validated as described in Section 5.
- All related documentation should be updated.
- The change should then be implemented.

Figure 3 gives a sample flow diagram of the procedure.

It may also be necessary to allow provision for the development, testing and implementation of approved emergency changes in the production version of the system. An expedited version of the change management procedure should be in place for these very exceptional circumstances.

Version control is essential for tracking all changes made to the programs and associated documentation to provide a complete history of the software and its various versions, as well as to ensure that the version of software and documentation in use at any given time can be uniquely identified and controlled.

Figure 3 Change Management Flow Chart



6.5 Periodic Review

To ensure that the validated production version of a system continues to perform as expected, its validation status should be reviewed periodically. It is advisable to check that the whole system is functioning correctly following a number of validated changes to its constituents. Moreover unnoticed environmental changes (e.g. to network software) may have occurred. A periodic review procedure should require such an activity to take place at least once every two years [6] if no other major validation activity has occurred meanwhile.

The level of review required should be based on a documented risk assessment. Any required testing could be based on an abbreviated version of the specific test data and methods detailed in the Test Plan which is designed to test the full functionality of the system. The review should be documented.

In addition, routine quality control procedures (*see Section 8.1*) and quality assurance audits (*see Section 8.2*) may detect system issues. These may lead to a 'for cause' review of the validated status of the system.

6.6 Decommissioning

When a system is to be decommissioned, a plan should be formulated which documents the necessary activities. These might include decisions on when use of the current system will be terminated, and which clinical studies will be completed on the current system and which will be transferred to the new system. The process of transferring clinical data from the current to the new system should be validated. Measures should also be in place to ensure that archived data from studies processed on a decommissioned system can still be accessed and read.



ONE-OFF AND STANDARD PROGRAMS

7. ONE-OFF AND STANDARD PROGRAMS

This section addresses a whole range of programs for the manipulation and/or analysis of data. They range from one-off programs developed for use with a single set of data to standard programs developed for repeated use with different studies (*see Section 9.7.2*). As with any software, these programs should be validated. Although the principles and rationale still apply, the scale of validation should reflect the size and complexity of the program, the extent of its use and the consequences of program error. Points to consider are that:

- the level of risk may vary according to the clinical development phase of the study.
- even if a program is intended to be a one-off, it should be documented sufficiently well to facilitate re-use if required
- a one-off program validated for a particular data structure cannot be assumed to be validated for a different data structure.

7.1 Responsibilities

Appropriately qualified and experienced personnel should be appointed to take responsibility for, and to perform, the validation. For the validation of one-off and standard programs, the developer and user are likely to be the same person. However there may be a significant additional risk in this person being solely responsible for the quality of the program. Management controls should be in place to minimise this additional risk, e.g. SOPs, programming standards (*see Appendix 8*).

7.2 Documentation

Documentation of the program (e.g. user requirements and specifications, usage, parameters, programming steps) can exist within the source code. The requirements and specifications may also appear within an overall list of programs used for the clinical study.

7.3 Software Standards

The software should be designed and programmed according to internal source code development standards, to ensure that quality source code is produced. Appendix 8 contains examples of programming standards.

7.4 Coding Review

The source code should be reviewed by someone other than the developer for accuracy and appropriateness of algorithms and formulae, clarity of organisation, internal documentation, change control and conformance to programming standards (see Appendix 8). If, due to the amount of source code, it is not practical to perform a detailed line-by-line review, then a high-level review using diagrams or charts as cognitive aids can verify the use of quality development practices. The review should be briefly documented, stating the date of the review, name of the reviewer and any non-conformance identified. A printout of the program code passing inspection should be signed and dated by the reviewer and archived, for example, with the study file. Whatever form a coding review takes, it should be performed in accordance with a written approved procedure.

7.5 Testing

Testing should be performed according to a pre-defined test plan, both during and on completion of the software development to identify and eliminate any errors in the programs and to provide confidence that the software is producing the desired output. Appendix 9 contains guidelines for testing one-off programs. The results of the tests, usually in the form of computer output, should be signed, dated and archived, for example, with the clinical study file.

7.6 Change Control

If any program needs to be modified following completion of the review and testing process, then this should be done with sufficient commenting/documentation within the program according to documented change control procedures (see Section 6.4). The reason for the change, the date of the change, the person implementing the change, and the new version number of the program after the change has been implemented should all be documented. To this end, it is recommended that the output should include the version number of the program, the date it was run and the person who ran it (see Section 9.4.8 and Appendix 8).

A formal validation report should not be required for one-off programs, however, a document should be signed to confirm that the validation procedures have been followed and the testing has been appropriate and consequently the software is authorised for use. A validation check list is an efficient method of tracking and documenting progress and ensuring that the appropriate tests are performed. It can also be used to record the authorisation for the use of the software and, therefore, may be considered as a validation report in this instance.

QUALITY MANAGEMENT

8. QUALITY MANAGEMENT

Quality management is a process involving people, systems and supporting tools and techniques that generates a product to the required specification. In this case the process is computer system validation and the product is a validated system. Quality management encompasses both Quality Control (QC) and Quality Assurance (QA) activities and has a role to play in achieving and maintaining the validated status of any computer system.

8.1 Quality Control (QC)

QC procedures are a series of routine prescribed activities performed at various stages of a process to ensure the quality of the end-product. Such activities can identify issues with any part of the system, i.e. hardware, software, people and processes. These in turn may lead to a 'for cause' review of the validated status of the system (*see Section 6.5*), revisions to SOPs and training materials and/or retraining of system users. Examples of QC activities within the computer system validation process include:

- regular review of user problem log
- regular check of access control levels
- review and approval of procedures and documentation
- review and approval of software.

8.2 Quality Assurance (QA)

QA is the structure provided by a quality system and would include all elements already referenced in the guideline plus independent audit and, therefore, this section will concentrate on the issues concerned with the audit of computer systems. The purpose of an independent audit of a process is to verify its compliance with SOPs and applicable regulatory guidelines, however, the compliance to SOPs and regulatory guidelines is the responsibility of user-management. Audits may focus on either how the system achieved validated status (i.e. how it was developed and implemented) or how the validated status has been maintained. The audits may be planned and scheduled to an agreed timetable or performed after a major change to the computer system (including people and processes).

Extracts from the ICH GCP and Statistics guidelines are given in Appendix 1. Other published guidelines and standards for system/software development and validation are listed in Appendix 10. The appropriate guidelines and standards to be audited against should be determined in advance and will depend on the system being audited. Audits should be performed on systems developed internally, commissioned systems, those purchased from vendors, those used by contract research organisations (CROs) to fulfil specific contractual obligations and investigator-owned systems.

8.2.1 In-House Audits

These audits should be performed against the current SOPs. Consequently the audit SOP should include an extensive checklist to assist the auditor. SOPs relating to computer system validation should be compliant with regulations and policies. Systems should also be in place to ensure that any changes to the regulations are quickly incorporated into the SOPs. The audit may identify suggestions for improvements as well as non-conformances.

8.2.2 External Audits

These audits should be conducted against a draft contract of agreed services and/or products prior to signing a contract or purchasing the software. The vendor or CRO should have SOPs covering all appropriate areas (*see Section 4*). SOPs and all other documentation should be under strict version control. The auditor should seek evidence that all documentation has been produced within a formal quality system, e.g. according to written procedures and produced by appropriately qualified personnel. The lack of a quality system, or any missing documentation, should be assessed against the risks to the customer's requirements. The financial stability of the CRO or vendor should be considered, to assess the likelihood that the system will be supported and maintained for a number of years, or that the service will continue for the duration of the contract.

8.2.2.1 CRO Audits (including Laboratories)

It is the responsibility of the customer to define their requirements before any contract is signed. The content of each audit will vary depending on the services being provided and the importance that the CRO's computer system plays in those services. The computer system should not be audited in isolation from the services being provided. The following steps are recommended to audit a CRO computer system and associated services:

1. Identify the services being provided, which SOPs should be used for those services and where these are documented.
2. List the risks to the customer's data associated with use of the CRO's computer system.
3. Determine the SOPs and other documentation necessary to address the risks, e.g. validation of one-off programs, change management for the computer system, security of data, etc. (*see Appendix 5 for a documentation checklist*).
4. Arrange an audit date and logistics.
5. Send an outline of the planned audit to the CRO.
6. During the audit, assess:
 - compliance with SOPs and documented requirements
 - the quality system of the CRO
 - the validation state of the computer system(s) used in provision of the services
 - maintenance level, calibration and service agreements, where appropriate.

7. Write an audit report, including any issues that need to be addressed.
8. Follow up any corrective actions until resolved, if necessary performing a follow-up audit.
9. Conduct periodic audits during the term of the contract to ensure ongoing compliance.

8.2.2.2 Vendor Audits

The following steps are recommended to establish the validity of third party software:

1. Identify the elements of the system, i.e. installation, operating environment, user training, software development, etc., and note who is responsible for each.
2. For the elements of the system under vendor control, list the risks to system validity.
3. Determine the documentation necessary to address the risks, e.g. minimum documentation for software development might be functional specification, design specification, test plans (including expected results), test results, system test report and sign-off (*see Appendix 5 for a documentation checklist*).
4. Arrange an audit date and logistics.
5. Send an outline of the planned audit to the vendor.
6. During the audit, assess:
 - the effectiveness of the vendor documentation in covering the identified risks
 - the quality system of the vendor
 - the validation state of the computer(s) used to develop the software particularly in relation to change control and backup, to ensure that the vendor can always determine/recall the exact version of the software being supplied.
7. Write an audit report, including a statement on the validity of the system, and identifying any issues that need to be addressed.
8. Follow up any corrective actions until resolved, performing a follow-up audit if necessary.
9. Conduct periodic audits during the term of the contract to ensure ongoing compliance.

Additional evidence may be sought to increase confidence in the system by extra testing or documentation at the customer's site prior to production use.

8.2.3 Investigational Site Audits

Computer systems of any sort used at the investigational site to collect data should also be validated. If the system is provided by the investigator, whether it is equipment which is used in normal medical practice and is to be used to collect trial data, or a system produced especially for the trial i.e. to enter data at the bedside, the same principles of validation apply. Where an investigator provides a computerised system to collect trial data the sponsor should consider the investigator to be a vendor.

If the system is supplied by the sponsor, the sponsor should have validated the system in advance so that the site audit could be limited to:

- the physical and logical security of the hardware and software
- the identification and training of staff using the system
- correct usage of the system
- maintenance procedures
- back-up procedures.

If the computer system used to collect data (e.g. remote data entry or 24 hour Holter monitor) is supplied by the investigator, the 'vendor' audit should be performed as early as possible during the planning of the trial and would assess:

- the effectiveness of the documentation to cover identified risks
- the quality system elements
- the validation state of the computer(s) used to develop the software
- the maintenance level, calibration and service agreements of collecting devices (i.e. 24 hour Holter monitor), if appropriate.

During the study, routine site audits should assess the same elements as for a sponsor supplied system and also:

- the continued maintenance level, calibration and service agreements of collecting devices, if appropriate
- the continued maintenance of the quality system elements
- procedures used to maintain the validated state of the computer system.

SPECIFIC CLINICAL SYSTEMS

9. SPECIFIC CLINICAL SYSTEMS

During validation of all clinical research computer systems, the general principles described in the previous sections should be considered and applied. This section describes some of the types of system found in the clinical environment and highlights areas to which specific attention should be paid.

9.1 Randomisation Systems

Randomisation systems, which are usually used by statisticians and/or pharmacy staff, may provide any of the following:

- a list of random numbers
- code-break envelopes
- packaging labels for drug supplies
- an electronic file of the patient treatment codes to be incorporated into the study database after it has been locked.

Validation of the randomisation system should be rigorous as randomisation codes and code-breaks, and their security, are key to maintaining the integrity of any clinical trial. The codes and code-breaks, generated prior to the start of the trial for the packaging of medications, will not be linked to the data until the end of the study when the clinical database has been locked, which may be several years after the codes were produced. The following points should be considered during validation.

- The source of the core random number generator and its validation status.
- The ability to reproduce the randomisation schedule. (This is a regulatory requirement [4].)
- Storage of randomisation codes and code-breaks and access control.
- Back-up and restoration procedures and their regular testing.
- Linkage to the Clinical Database Management System (*see Section 9.4*), Administrative System (*see Section 9.13*) and Drug Supplies Accountability System (*see Section 9.6*).

9.2 Data Capture Systems

9.2.1 Automatic Measuring Devices

Automatic measuring devices are medical devices and so must comply with regulatory guidelines, e.g. the FDA Medical Device Quality System Regulation [7]. This covers design controls which include requirements for the validation of software used in such devices. Draft guidance has since been issued which further addresses the validation of medical device software and the software used to design, develop or manufacture

such software [8]. Automatic capture of data from clinical measuring devices (e.g. EEG equipment, Holter devices) should address the following specific validation issues:

- auditing of the vendors of any sponsor-supplied equipment
- assessment of the investigator site prior to the start of the trial, including investigator-supplied equipment considering such issues as calibration and maintenance procedures (*see Section 8.2.3*)
- validation of pre-processing software prior to transfer to the clinical database management system (CDMS)
- all requirements associated with electronic data transfer (*see Section 9.3*) between systems and/or between sites.

9.2.2 Manual Data Input

9.2.2.1 In-House Data Entry Systems

These may be stand-alone data entry systems or the data entry functions of the CDMS, which provide the facility for manual key entry of clinical trial data from paper case report forms (CRFs) and other paper sources, e.g. diary cards.

Specific considerations for the validation of manual input from paper copy include:

- input screen testing, i.e. the fields should be correctly defined to the system
- documentation and functional testing of any data checking routines that execute upon entry of a data item, e.g. range, date, format, coding, field dependencies
- testing of any entry verification functionality, e.g. on-line second entry verification, file comparison, batch verification
- batch transfer to the clinical trial database from separate data entry systems (*see Section 9.3*).

9.2.2.2 Remote Data Entry Systems

Remote data entry systems are systems deployed at the investigator site for direct manual input either during or after patient assessment. These systems will often incorporate data checking routines to expedite data query resolution.

Specific considerations for the validation of remote data entry systems are:

- specific user requirements
- the personnel who will enter the data, e.g. clinician, nurse, clinical data manager
- timing of data entry relative to patient consultation
- vendor auditing (*see Section 8.2.2.2*)
- validation of the remote systems (*see Section 9.2.2.1*).
- validation of any incorporated data checking programs (*see Section 9.4.4*)
- audit trail of changes to entered data
- validation of the transfer of remotely captured data to the host database (*see Section 9.3*)
- security of the transfer medium, e.g. telephone lines, Internet.

Given the environmental conditions of the remote data entry system, aspects which may present particular difficulties in the maintenance of the validated state are:

- training of personnel both internal and external to the sponsor company
- Help Desk support for remote site users
- access to, and security of, the installed software and the captured data (where supplied specifically for a clinical trial, a computer should be used for this alone: where part of the computer system is used by the investigator for other purposes, there should be logical and physical isolation of the clinical trial software to preclude interaction with non-study software)
- back-up and restoration procedures.

In situations where the system is not supplied by the sponsor company, the validation of these systems is beyond the control of the sponsor. In these circumstances, availability or evidence of the following items should be considered as a minimum before using such systems:

- an adequate system description
- test documentation
- configuration management
- changes or enhancements to the system performed in a controlled manner
- effective procedures for using the system
- existence of system management procedures.

9.2.2.3 Electronic Diary Cards

These portable, hand-held systems designed to be programmed according to specific protocol requirements are used by patients to record directly information on their condition and/or medication consumption during a particular study. They should be specified and designed so that they are highly prescriptive since they are used in a relatively uncontrolled environment (e.g. patient's home). Specific considerations for the validation of electronic diary cards are:

- suitability for use by the target patient population
- functionality, e.g. time of data capture, checks for logical consistency, data confirmation and auditability, provision of investigator signature
- vendor auditing (*see Section 8.2.2.2*)
- usability, robustness and integrity of both software and hardware
- tamper-proof software, i.e. modification for other purposes should not be possible
- power back-up in the event of expiry or removal of batteries
- security, controlled by password, including access restrictions and integrity of data
- transfer of the diary data to the host database, including any data modification, annotation or processing occurring before, during or after the transfer (*see Section 9.3*)
- documented training of site personnel and individual patients.

9.2.3 Automated Data Input

9.2.3.1 Optical Character Recognition (OCR) and Intelligent Character Recognition (ICR)

OCR/ICR systems recognise images as alpha-numeric data, as if the data had been entered directly from a keyboard. They do this via recognition engines, operating by template matching, feature extraction, neural networking or a combination of these approaches.

There is an explicit reliance on operator involvement in the verification of the captured data, whereby the software presents the operator with uninterpretable input image for manual resolution. Validation needs to take account of all dimensions of the system, testing with a sufficiently varied selection of input image. Specific considerations for the validation of OCR/ICR systems are:

- vendor auditing (*see Section 8.2.2.2*)
- reliability, calibration and maintenance of scanners
- correct identification by the system of the type and number of scanned input forms
- functionality, e.g. substitution, learning capacity, verification
- reliability of interpretation of the specified field images by the recognition engine
- handling of indeterminate data
- training and competency of the operator
- transfer to the host CDMS or analysis package (*see Section 9.3*)

9.2.3.2 Bar Coding Systems

A bar code is a pattern of dark bars separated by spaces. The bar code is read by passing a beam of light over it. Light is absorbed by the bars and reflected by the spaces. The differences in reflection are sensed by the scanning device (e.g. light pen, hand held scanner, flat bed scanner) and converted into electrical signals corresponding to the widths of the bars and spaces which can then be decoded into the numbers and letters represented by the bar code. There are a number of different bar coding standards.

Specific considerations for the validation of bar coding systems are:

- the system used for creating the bar code labels, e.g. acquisition of number, conversion of number to bar code
- print quality of the bar code, e.g. specks of ink in the spaces, edge definition of the bars, and print contrast between the bars and spaces
- presentation of the bar code to the scanner, e.g. creased labels, protective covering
- robustness and maintenance of the scanning device
- verification of decoding
- control of the re-use of pre-printed bar-codes.

9.3 Electronic Transfer of Data and/or Software

Clinical data and/or software may be transferred electronically, by diskette or direct

line, on a routine basis from investigator sites, contract research organisations or central laboratories to the company (and vice versa), between different company locations, between computer systems within a location, and from the company to regulatory agencies.

Specific considerations for the validation of electronic transfers are:

- Internet, intranet and/or other communication technologies (e.g. group ware, modem-to-modem, cellular technology)
- externally owned lines
- communication medium (e.g. diskette)
- security (e.g. encryption, passwords, virus protection, 'fire walls')
- specifications of the transfer file:
 - file format (e.g. ASCII, comma delimited)
 - size of file
 - number of records
 - linkage of comments to numeric data
- recovery following interruption of transmission
- corruption during transfer
- consistency of electronic file with source
- back-up and disaster recovery in both the sender and receiver locations

9.4 Clinical Database Management Systems (CDMS)

A clinical database management system usually comprises a combination of sub-systems enabling the user to carry out a wide variety of tasks. The management and use of the system, and related reference data (e.g. laboratory reference ranges and coding dictionaries), should be controlled by SOPs.

The validation of a CDMS should be carried out according to the principles outlined in Sections 1-8 of this guideline. The nature of the validation will depend on the system's complexity and its source (*see Section 5*).

Points to consider during the validation of some aspects of a CDMS follow.

9.4.1 Database Set-up

At the system level:

- Libraries of templates (e.g. standard data set structures) should be accessible.
- Templates should be adaptable for specific studies.
- Study-specific requirements should be possible.

At the study level, it should be possible to set up the database efficiently to allow easy access to the data. Where the functionality is available, the set-up should be tested for each study using dummy data to ensure that:

- entry screens function as expected (e.g. range checks, look-up tables, auto-encoding using the appropriate dictionary, derived data calculations)

- entry screen fields correctly relate to database fields
- fields are correctly defined in terms of format (e.g. character/numeric, length)
- entry of confirmed missing values is possible.

9.4.2 Data Capture

The validation of various data capture systems is described in Section 9.2. However the transfer of data from such capture systems into a CDMS should also be addressed (*see Section 9.3*). Points to consider here include:

- Verification, if part of the transfer process, should result in discrepancies between two manual entries being correctly identified, and their subsequent resolution should result in one correct entry on the database.
- The transfer process should enable single entry of certain data, e.g. electronic laboratory data.
- Data should be loaded into the correct location, i.e. table and field.
- The transfer process should detect duplicate records.
- The user should be notified of non-transferred data.
- Any data identifiers should be correctly assigned.
- The date and time of initial loading of each data item to the database should be recorded by the system, i.e. the audit trail should commence at initial loading (*see Section 9.4.5*).
- Any auto-encoding, if part of the transfer process, should function as expected using the correct dictionary for each coded variable.

9.4.3 Derived Data

Any data derived within a CDMS should be validated (*see Section 9.5*).

9.4.4 Data Checking

Data checking may also be referred to as edit checking, plausibility checking, range and consistency checking or data validation. At the system level points to consider include:

- Libraries of standard data checks should be accessible
- Standard data checks should be adaptable for specific needs
- Study-specific data checks should be possible
- Study-specific checks should be correctly incorporated, with standard checks, into the study-specific editing functionality
- The checks should be executed correctly, i.e. the correct checks should be applied to a data item at the appropriate time
- Data items accepted following a failed data check should not fail again unless the data change
- Failed data checks should remain flagged until resolved.

At the study level the set of specified edit checks should be tested using tailored dummy data to ensure the absence of false positive and false negative failures.

9.4.5 Audit Trail

Validation of the audit trail facility should address the following functionality:

- Date and time stamping of the initial loading of a data item and all subsequent changes, plus the identification of who entered the data and who subsequently made changes and, if possible, the documented reasons for change.
- The ability to reconstruct the database as it existed at any date and time in the past.
- Protection of the audit trail such that no direct modification of the stored information may be made, but read access is possible at any time.

9.4.6 Database Snapshot

Validation should address the ability of the system to produce a time frozen representation of the database, i.e. to ensure that it is an accurate representation at that time point.

9.4.7 Data Review

Any process which involves review of clinical data relies on its accurate presentation. Validation of software used to produce reports from a CDMS should follow the guidelines in Section 7. Computer-aided review tools are covered in Section 9.8. Points to consider include:

- extraction of the data for the correct study
- extraction of data from the correct data tables, individually or merged
- extraction of the correct data items in the correct format.

9.4.8 Reporting

At the system level the functionality of a reporting system should ensure that:

- template programs are available for easy adaptation
- study-specific programs can be easily developed
- program development takes place in a separate environment from the use of validated programs
- documentation of output should include source program, date and time generated, user, page number and total number of pages (*see Appendix 8 and Appendix 9*).

All programs, and sub-routines or macros called within programs, used to produce formatted output from the CDMS for clinical reports should be validated (*see Section 7*).

9.4.9 Database Locking/Securing

Validation should ensure that the facility for locking/securing the database prevents unauthorised write access. Unlocking of a database should be strictly controlled by an SOP.

9.4.10 Archiving and Retrieval

The database plus any pertinent reference data sets, e.g. laboratory reference ranges and coding dictionaries, should be archived in a format (e.g. ASCII) which enables them to be reloaded into the required format at any time in the future by the same

CDMS, or a new one if the original has been decommissioned. The retrieval and restoration of data should allow processing to continue.

9.5 Derived Data

Derived data algorithms are the programmed implementation of formulae used to derive new variables from variables for which data have been collected. Examples include:

- Age = date of a specific time point within the study schedule (e.g. date of randomisation) minus date of birth, rounded down to the nearest year.
- Conversion of laboratory data from the units in which they were observed to specified standard units for reporting purposes.
- Derivation of a laboratory data flag (e.g. High, Normal, Low) based on the relationship of the observed value to the appropriate laboratory reference range, which may depend on demographic data.
- Pharmacokinetic parameters.
- Derivation of adverse event preferred terms by auto-encoding verbatim text against an adverse event dictionary.
- Transformation via table look-up.
- Patient evaluability-for-efficacy status.

The software developed to create the data for the new variables may be applied during data entry, loading to the database or analysis/reporting. In the first two cases, the data for the new variables reside within the database. In the latter case the values of the new data are only evident from the output of the analysis/reporting programmes.

Issues to be considered in validation are:

- adequacy of algorithm specification
- algorithm version control (so that the same version can be used across studies within a development programme)
- impact of algorithm re-specification on historical derived data
- facility for manual over-ride
- coding review (*see Section 7.4*)
- testing with extreme, boundary and missing values of the source variables, either alone or in combination
- testing all routes of calculation, where the route depends on values of the source variables
- use of in-built checks on the derived data values (*see Section 9.4.4*)
- translation of source variable names to their data entry or database names
- dictionary and look-up table version control.

9.6 Drug Supplies Accountability Systems

These systems record whether or not all dispensed medication for a clinical trial can be accounted for at the end of the study. For each subject in the trial the amount(s) of

dispensed medication are compared with the amount(s) consumed and the amount(s) returned. The returned supplies are then destroyed and certified as such. The amount(s) dispensed may come from a pharmacy system and the percentage consumption within a dispensing interval could be derived as a measure of subject compliance and transferred to the CDMS. Specific points to consider during validation are therefore:

- The incorporation of any derived data algorithms (*see Section 9.5*).
- Electronic transfer from and to other systems (*see Section 9.3*).

9.7 Statistical Systems

The statistical software systems used for analysis of clinical trial data can range from custom programs for specific statistical techniques to commercially available packages. Such packages (e.g. the SAS system, SPSS, S-Plus) provide the user with a library of statistical procedures (e.g. analysis of variance, regression, generalised linear modelling, non-parametric methods) which can be accessed either by using the native programming language, or by selecting the required options from the package's user interface.

9.7.1 Commercial Systems

It is generally considered that there is no requirement for validation of statistical packages such as the SAS system as entities in their own right. Nevertheless any custom program written using the package's native programming language should be validated (*see Section 9.7.2*).

The vendor-supplied installation tests should be performed and documented to ensure that the software is functioning correctly within the specific operating environment. In addition, a suite of vendor supplied programs, test data and results /output can be a valuable aid to validation. Repetition of all these tests should be considered each time there is a change to the operating environment (*see Section 6.4*).

9.7.2 In-House and Commissioned Systems

These include one-off and standard programs and macros (*see Section 7*) developed using either a non-statistical programming language (e.g. Fortran) or the native programming language of a commercial statistical software package (e.g. SAS programs, SAS macros, SAS/AF applications).

It should be shown that statistical procedures and functions (e.g. SAS PROCs), supplied as part of a commercial package, are used correctly within the context of the program. Software which automates the data analysis process across a number of clinical trials should be validated in the same way as other in-house or commissioned systems (*see Sections 5.1.1 and 5.1.3*). However the validation requirements for trial-specific, one-off programs written using commercial package native languages are reduced (*see Section 7*).

Specific issues to consider during validation are:

- statistical competency of the developer
- precision and rounding errors
- handling of missing data values

- handling of unequal (unbalanced) treatment groups
- handling of ties in non-parametric analyses
- facilities for checking the underlying assumptions of the statistical model
- facilities for excluding outlying observations from analysis
- printing of intermediate values during calculations
- statistical competency and training of the users
- operating environment and conditions.

9.8 Computer Aided Review Tools

Validation of computer aided review tools, which may be used by in-house or regulatory reviewers to explore the project database on a read-only basis, should address both the system and project-specific aspects.

The underlying code of the generic shell that comprises the tool should be developed according to the Software Development Life Cycle. Depending on the degree of sophistication of the system, testing should cover the following areas of functionality:

- selection of compound
- selection of trial
- display of raw data
- subsetting of data
- 'point and click' cascading menus (i.e. increasing or decreasing the level of detail or sub-setting)
- search facilities
- display of graphical results
- linkage between annotation facilities (for sponsor and/or reviewer) and related data
- transfer of data between different softwares (e.g. from the SAS package to a spreadsheet)
- analysis and reporting.

Testing for correct project and study set up should demonstrate that the data have been loaded into the system correctly. This will involve the checking of meta-data, for example:

- completeness, correctness and consistency of the labels and formatting
- correct functioning of the screens
- consistency of the viewed data with the project database
- consistency of reports and views of data output to the screen with clinical trial report tables, listings and original CRFs.

The different ways of viewing data may be too numerous to test exhaustively. Validation requirements therefore need to be realistic to ensure an appropriate level of overall confidence.

9.9 Electronic Document Management Systems (EDMS)

Document management refers to procedures or systems designed to exert an intelligent control over the creation, management and distribution of documents. Electronic document management systems (EDMS) may include any or all of the following features:

- controlled document authoring
- electronic storage of documents or data, either scanned in (*see Section 9.9.1*) or created electronically (*see Section 9.9.2*)
- controlled distribution of documents to, and retrieval by, multiple users
- review and/or approval of documents, e.g. within a work flow component (*see Section 9.9.4*)
- publishing of approved documents
- archiving of documents for completed projects (*see Section 9.9.5*).

Typically, EDMS may need to address a range of issues including version control, access control, organisation and management, workflow, imaging, publishing, document re-use, indexing and searching.

The document types that may be stored within the system may have a wide variety of file formats and sources and range from just key documents to the totality of documents generated for a project. For each type of document stored in the EDMS, it is important to define the 'original' version (i.e. as paper or electronic).

The validation issues for EDMS include:

- the life-span and characteristics of the storage medium used, including the frequency and type of testing required
- the security levels of the documents and the system, including process-specific security such as that used for electronic signatures (*see Section 9.10*)
- version control of documents including audit trail
- continuing readability of documents through technological changes, e.g. the use of Portable Document Format (PDF) file type
- validity period of printed/published documents, e.g. SOPs
- the qualifications, training and competency of users
- indexing functionality.

9.9.1 Optical Imaging

Optical images may be produced by scanning in a paper document or a faxed image into the system. Apart from general configuration and installation requirements, specific validation considerations should include:

- procedures for calibration and maintenance of scanners
- definition of the master record, i.e. paper version or electronic image
- routing of images to appropriate locations
- interfaces with other systems
- for fax-to-image, correction of transmission errors

- readability of retrieved images
- image quality prior to destruction of the original document
- indexing functionality.

9.9.2 Electronic Document Repository

A fundamental characteristic of most EDMS will be some form of document repository. This will generally be a clearly defined and secure area for the storage of all documents associated with the EDMS. The repository may range from a specific directory on a server, with a work group password protection, to a software package controlled database repository implementing full database security controls. There may also be a requirement to produce and store multiple renditions of a document within the repository.

Specific validation issues include:

- continuing readability following software upgrades
- integrity of the document during conversion
- production, storage and retrieval of multiple renditions of a document
- storage of signatures associated with documents (*see Section 9.10*).

9.9.2.1 Electronic SOPs

An electronic document repository may be used to store SOPs, allowing easy and ready access for all staff. The repository may include the following features:

- workflow for approval of SOPs
- electronic records and signatures (*see Section 9.10*)
- facility for user requests to change an SOP
- storage of forms and templates in original software
- storage of previous, current and under-revision versions
- controlled distribution.

Specific validation issues include:

- access security, especially write access to approved SOPs
- documentation of notification of new/revised SOPs to all appropriate staff
- version control
- integrity of the system, especially when replicated across servers
- control of printed versions of SOPs.

9.9.3 Indexing

Document indexing may range from simple attribute indexing through to full document text indexing. Specific validation considerations include:

- manual entry of index keys (*see Section 9.2.2.1*) including verification (e.g. second entry, plausibility checking)
- automated entry of index keys through OCR/ICR or bar coding (*see Sections 9.2.3.1 and 9.2.3.2*)

- index generation from the input index keys (e.g. integrity, error detection, duplicate identification)
- retrievability of indexed documents.

9.9.4 Work Flow Systems

Work flow systems can be used for many purposes such as:

- development and approval of protocols and clinical study reports and their amendments
- routing CRFs through medical review and the clinical data management process.

Issues to be considered during validation include:

- testing of all possible routes to ensure that a document does not become suspended within the system
- testing of parallel tasks to ensure that the result of those tasks is the same regardless of their sequence in real time
- linkage and preservation of electronic annotations
- corruption of the master document by annotations
- printing of document and annotations.

9.9.5 Archiving

An important part of any electronic archive system is a policy or SOP, which will affect the validation effort, including:

- which documents should be kept in hard copy form and which may be kept only in their electronic form
- how long documents are maintained on the system
- how long hard copies of documents are kept
- the need for off-site electronic back-up when hard copies of documents are destroyed
- the possible uses of the documents, including whether they may be required by a court of law
- access to the archive
- storage criteria for electronic media and any special considerations e.g. refreshing tapes/disks.

9.9.6 Document Sign-off/Authorisation

In most cases there will be a requirement for a document to be appropriately approved and signed-off. A particular issue for EDMS is how this is captured and stored. Options include the scanning and storage of the signed document, scanning and storage of the signature(s) associated with an electronic document and the use of electronic signatures (*see Section 9.10*)

The validation issues include:

- signature verification and security, including fraud detection
- storage of signatures associated with specific documents.

9.10 Electronic Records and Electronic Signatures

The regulation aims to define the circumstances in which electronic records will be considered by the agency to be equivalent to paper records, and electronic signatures to be equivalent to traditional handwritten ones. Although originating from discussions surrounding GMP-based applications, the scope of the regulation covers all records required by the agency and also those records contributing indirectly to any submission. This includes clinical data.

This rule has several potential implications on the design and operation of clinical computer systems, and, therefore, will impact validation. Amongst the new requirements within the rule, the key elements are:

- Certification to the agency by persons using electronic signatures that they intend them to be legally binding.
- Written policies which hold individuals accountable for actions initiated under their electronic signatures.
- Electronic signature records which include a printable/readable name of the signer together with the date and time of signature.
- Electronic signature records which are permanently associated with their relevant action(s) where the extent of this action and the reason for it are clear.
- The ability to generate a human-readable electronic signature record, e.g. a print out.
- Where biometric/behavioural links are not in use, two distinct identification mechanisms to confirm a signing action, e.g. id-code and password.
- Identification mechanisms which are uniquely associated with an individual user.
- Electronic audit trails for all operator activities which create, modify or delete records.
- The significance of system security procedures and controls that limit access to authorised individuals so that the integrity of electronic data is promoted throughout the whole period of system use.

It is not easy to create a regulation that is equally applicable to key process milestones such as clinical database closure as well as to everyday electronic data capture activities without compromising the latter's productivity. However the rule does not demand that companies must implement electronic signature systems nor that records must be submitted electronically. The use of such technologies remains voluntary.

The Final Rule became effective on 20th August 1997. Its requirements should therefore be taken into consideration during the specification of the record handling, security and authorisation functions of any new system, including its data archive. Once included in the specification the functionality associated with electronic signatures and electronic records should be part of the application's Acceptance Test. Existing systems should be assessed against the proposed rule and shortfalls identified and assessed. If these shortfalls could bring the integrity of trial data into question, system modification should be considered.

It should be noted that although the above comments refer specifically to the FDA

regulation, similar directives and guidelines have either been published recently or may be anticipated from other agencies.

9.11 Electronic Publishing and Regulatory Submissions

Electronic regulatory submissions combine components from specific systems, e.g. computer aided review tools (*see Section 9.8*) and electronic document management systems (*see Section 9.9*). Electronic publishing systems assemble electronic documents and images into electronic dossiers. The validation requirements of the publishing system, over and above the requirements for each component system should be assessed.

9.12 Pharmacovigilance Systems

Pharmacovigilance systems capture, store, process, maintain, classify and report adverse event data. Any such systems generating reports for regulatory authorities (e.g. expedited reports, periodic safety updates) and the interfaces into them from a variety of sources, should be validated. Specific considerations when validating these systems are:

- reconciliation of adverse event data from the clinical trial database, and other sources, with the pharmacovigilance database through electronic interfaces
- development of programs to generate reports for regulatory authorities, e.g. expedited and periodic reports
- assurance that all cases known to the system have been appropriately reported in the appropriate time frame
- electronic transfer to regulatory authorities (*see Section 9.3*).

9.13 Administrative Systems

Clinical administrative systems are used to manage clinical development programs from first study in man through to regulatory submission and beyond. Functionalities can include the following:

- Master Repository - for descriptive information (e.g. names, codes) about projects, studies, investigational sites, test treatments, etc.
- Project Management - for the maintenance of time lines. Process templates, which identify tasks, milestones, linkages between them and task durations, can be adapted for individual studies.
- Resource Management - for the allocation of manpower resources to tasks and generation of work schedules.
- Site Monitoring - for tracking subject enrolment, visit reports, investigator payments, etc.
- Administrative Tracking - for monitoring the status of the flow of case report forms and associated queries through the process from retrieval to final database.
- Project Accounting - for the generation of historical reports and forecasts based on actual and planned resource allocation.

- Benchmarking - for the use of historical performance data (e.g. actual durations, intervals between milestones) for external comparisons or as a basis for evaluating internal process changes.

Although administrative systems are not required to be validated by regulatory guidelines, their complexity and the potential impact they can have on a clinical research organisation requires them to be developed according to established good practice, including the provision of documented evidence of effective controls. The approach to validation will depend on the source of the system, i.e. developed in-house, commercially available or commissioned (*see Section 5*).

Specific considerations in the validation of administrative systems include:

- write access for different individuals to different parts of the system
- the impact of a delayed actual date on subsequent plan dates, e.g. functionalities of automatic rescheduling, or alert mechanisms to those responsible for subsequent tasks (project management)
- electronic linkage between the administrative system and other systems which may depend on it for consistent descriptive data (*see Section 9.3*), e.g. randomisation systems (*see Section 9.1*), data capture systems (*see Section 9.2*), CDMS (*see Section 9.4*), drug supplies accountability systems (*see Section 9.6*), EDMS (*see Section 9.9*)
- electronic transfer of data from other systems, e.g. CDMS for administrative tracking (*see Sections 9.3 and 9.4*)
- derivation of summaries for status reports and historical reports for benchmarking and project accounting (*see section 9.5*).

GLOSSARY

10. GLOSSARY

Archive System - A system which copies data, often in a compressed form, and saves it onto tape or a different disk, either as a back-up or to allow access to that particular data at some time in the future.

Audit - An independent examination which includes the evaluation of systems and processes established to ensure that clinical trials are performed and data are generated, documented and reported in compliance with SOPs, GCP and applicable regulatory requirements.

Audit Trail - A function of a system, e.g. CDMS, whereby the date and time of all modifications to each data item, together with the identity of the person making the change and (not essential) the reason for the change, are recorded by the system. This allows the database to be recreated as it existed at any point in time.

Automated Testing - The use of software specifically designed to test certain aspects of a computer system, e.g. the amount of source code used, re-testing following system changes, regression testing (i.e. comparison of current and previous runs).

Automatic Measuring Device - A device (e.g. EEG equipment, Holter monitors) which accept biological input (e.g. blood samples, heart beats) to generate electronic source data.

Bar Code - A code which represents characters by sets of parallel bars of varying thickness, and separations, that are read optically by transverse scanning.

Boundary Values - Values that correspond to maximum and minimum input, internal, or output values specified for a system. They may be the maximum and minimum values themselves or values which lie just inside or just outside a specified range of valid input and output values.

CDMS - See Clinical Database Management System

Cellular Technology - Wireless communications

Change Control - The procedures employed to manage changes in the hardware, software, personnel or procedures of a system.

Clinical Database Management System (CDMS) - A system for clinical trial data entry, verification, checking, correction, transformation, manipulation, storage, retrieval, review and reporting.

Computer Aided Review Tool - A system for the exploration of a database, with read-only access, allowing the user to browse and select data for extraction, manipulation, analysis and reporting.

Computer Program - A collection of logically interrelated statements or instructions that when executed by a computer make possible the performance of a predefined task.

Computer System - The set of hardware, software, procedures and personnel which together perform a specific function or group of functions.

Data Capture System - A system for entry of data to a target system from paper source, an automatic measuring device, direct entry or electronic transfer.

Data Check - Comparison of a data point against a range of possible values, and/or other data point(s), to assess its validity.

Derived Data - Data items formed by manipulation of raw (source) data.

Derived Data Algorithm - The formula used to create derived data from raw (source) data.

Design Specification - A specification that documents how a system is to be built. It typically includes system or component structure, algorithms, control logic, data structures, data set (file) use information, input/output formats, interface descriptions, etc.

Development Environment - A computer environment for developing a system, which is sufficiently separate from the testing and production environments so as not to affect or be affected by them.

Dictionary - A list of allowable values for a specific data item.

Documentation - Any kind of written material, e.g. manuals, procedures or policies, records, or reports, which provides information concerning the use, maintenance, or validation of a computer system.

Drug Accountability System - A system for accounting for all medication dispensed within a clinical trial as either consumed, returned or misappropriated.

Electronic Diary Card - An electronic record into which a subject participating in a clinical trial directly enters observations or directly responds to an evaluation checklist.

Electronic Document Management System (EDMS) - A system for the controlled development, review, approval, distribution, storage, archiving and publishing of either electronic documents or scanned images.

Electronic Publishing System - A system that assembles components (e.g. electronic documents, images, data listings, analysis output) from different systems (e.g. Computer Aided Review Tools and Electronic Document Management Systems) into one or more paginated volumes with tables of contents and headers/footers.

Electronic Regulatory Submission - The submission of a marketing application by electronic transfer to the regulatory authorities of the output from the sponsor's Electronic Publishing System.

Electronic Signature - A computer data compilation of any symbol or series of symbols, executed, adopted, or authorised by an individual to be the legally binding equivalent of the individual's hand-written signature.

Electronic Transfer - The transfer of electronic data from one system to another via diskette, telephone, the Internet, etc.

End-User - The person, or persons, who operate or interact directly with the computer system.

Enhancement - A change to an existing computer application which adds to the functionality beyond previous specifications.

Escrow - A legal term in Anglo-American law. A written agreement, constituting evidence between two or more parties (in this case, vendor and purchaser), that is given to a third party with instructions (in this case, to deliver the source code) to be executed only upon a future condition (in this case, the vendor going into receivership).

Evaluation Plan - A complete plan as to how the process of evaluation of a Legacy System is to be achieved. All individual stages/ processes must be defined, testing process detailed and output documentation identified.

Evaluation Report - Report of the performance of the Evaluation Plan for a Legacy System.

Fire Wall - A system which screens, monitors and protects a computer system from unauthorised external access.

Functional Specification - Detailed specifications of the nature and structure of the system under evaluation, including the operating system.

Functional Testing - Testing conducted to evaluate the compliance of a system with specified functional requirements. It therefore ignores the internal mechanism or structure of a system and focuses on the outputs generated in response to selected inputs and execution conditions.

Groupware - A system of communication between members of a 'group' usually including such functionality as electronic mail, filing, personal diaries, meeting scheduling, task scheduling, notices, and message forwarding.

Indexing - The unique identification of electronic documents to allow filing, searching and retrieval.

Intelligent Character Recognition (ICR) - An information processing technology that converts human readable data into another medium for computer input. An ICR peripheral device accepts a hand-written document as input, to identify the characters from the light that is reflected by their shape, and creates an output disk file.

Legacy System - A computer system in production use with no validation documentation.

Meta-Data - Data which describe data, i.e. attributes of data storage variables such as name, type, format, location.

Neural Network - A hardware and software architecture which copies the human brain in form and structure in an attempt to improve the computer architecture paradigm and recognise spatial and logical patterns. Neural networks emulate the learning aspect of the human brain and are 'trained' to provide the correct response.

One-off Program - A program used with a specific set of data from a single study

Optical Character Recognition (OCR) - An information processing technology that converts human readable data into another medium for computer input. An OCR peripheral device accepts a printed document as input, to identify the characters from the light that is reflected by their shape, and creates an output disk file.

Optical Imaging - A system that scans a paper document or faxed image into a computer for retrieval as an electronic image.

Palm Recognition - The identification of a person based on a map of the palm. Similar in principle to traditional fingerprinting, palm recognition is based on optical scanning. Use of the complete palm rather than a single index finger improves confidence in the result.

Personal Identification Device (PID) - A token, typically the size of a credit card or pen, carried by a person. The device must be inserted into a PID reader connected to the user's workstation or terminal in order to gain access to the system. PIDs are usually allocated one per person and the system will ensure that the PID correctly matches against the user name and password keyed on the screen.

Pharmacovigilance System - A system that captures and handles those case histories of individual drug adverse experiences, arising from both clinical trials and marketing, which need to be reported to regulatory authorities within specified time frames.

Production Environment - A production computer environment, which is sufficiently separate from the development and testing environments so as not to affect or be affected by them.

Programming Standards - Rules/standards used when writing the code for a piece of software.

Prototyping - Using software tools to accelerate the software development process by facilitating the identification of required functionality during the analysis and design phases.

Quality Assurance - The planned systematic activities necessary to ensure that a system conforms to established guidelines, SOPs, standards and technical requirements.

Quality Control - The operational techniques and procedures used to achieve quality requirements.

Quality Management - A process implemented by a quality system which controls the delivery of a product assured to meet its specification.

Quality System - The organisational structure, responsibilities, procedures, processes and resources for implementing quality management (ISO 8402).

Rapid Application Development (RAD) - A structured software requirements discovery technique which emphasises generating prototypes early in the development process to permit early feedback and analysis in support of the development process.

Randomisation System - A system that generates the random allocations of treatment to successive patients within a clinical trial. Other functionalities may include generation of the associated code-break envelopes, and an interface to transfer the treatment codes to a CDMS after the database has been frozen so that the data can be analysed.

Raw Data - The initial recording of data or an exact copy of the initial data prior to any manipulation.

Release Notes - A document issued to all end-users at the time of release, identifying the system and describing or referencing a description of its functionality and known problems.

Retinal Scanning - Retinal scanning identifies a person based on the fact that no two persons have the same pattern of blood vessels on their retina, i.e. the tissue lining the rear wall of the eye.

Retrospective Evaluation - The evaluation of historical documentation on the development and use of an established computer system, and the resulting identification of any validation deficiencies, which need to be addressed to provide the needed formal assurance that the system does what it purports to do.

Security - The protection of computer hardware and software from accidental or malicious access, use, modification, destruction or disclosure. Security also pertains to personnel, data, communications and protection of computer installations.

Service Level Agreement - An agreement describing the level of system support and response time which will be provided to users by the support team after system implementation.

Software - A collection of programs, routines and subroutines that controls the operation of a computer or a computer system.

Software Development Life Cycle (SDLC) - The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes installing and checking out the software for operational activities. These activities may overlap or be performed iteratively.

SOP - Standard Operating Procedure: written details of how procedures should be carried out and who is responsible.

Source Code - An original computer program expressed in human-readable form (programming language), which must be translated into machine-readable form before it can be executed by the computer.

Source Data - All information in original records and certified copies of original records of clinical findings, observations, or other activities in a clinical trial necessary for the reconstruction and evaluation of the trial. Source data are contained in source documents (original or certified copies).

Structural Analysis - An analysis that identifies modules or other entities in a system and shows how larger or more general entities break down into smaller, more specific entries.

Test Environment - A computer environment for testing a system, which is sufficiently separate from the development and production environments so as not to affect or be affected by them.

Test Plan - A document prescribing the approach to be taken for intended testing activities, i.e. items to be tested, the testing to be performed, test schedules, personnel requirements, evaluation criteria, any risks requiring contingency planning and the documentation to be produced.

Testing - The process of exercising or evaluating a system or system component by manual or automated means to verify that it satisfies specified requirements or to identify differences between expected and actual results.

Thread Testing - Testing of a process from beginning to end, checking the outputs at each intermediate step.

Unit Testing - Testing of a module for typographic, syntactic, and logical errors, for correct implementation of its design and for satisfaction of its requirements.

User - See End-User.

User Acceptance Testing - Testing of software by the end-users against the User Requirements and Functional Specifications at the end of development and prior to release for production use.

User Management - The managers of the end-users. User management will typically hold executive responsibility for the introduction and operation of a computer system.

Validation Plan - A document describing measures and responsibilities agreed, assumptions about and limitations on the scope of the validation exercise, and justification for any exclusions.

Validation Report - A document which describes the outcome of executing a validation plan, and records the validation status of a computer system.

Verification (data entry) - A process of checking the first entry of data for typographical accuracy. One such process is second entry verification where errors are detected and corrected by the second operator at the time of second entry. Another is dual entry and file match, where the two independent entries are compared, and differences resolved, by a third operator.

Verification (software) - The demonstration of consistency, completeness and correctness of the software at each stage and between each stage of the development life cycle.

Version Control - Identification of each successive version of a document/system, by a sequential version number and date, and recording of the change history. This allows the development of the document/system to be followed, easy identification of the current version and which document/system was valid on any particular date.

Voice Recognition - The identification of spoken words by a machine. The words are digitised, i.e. turned into a sequence of numbers, and matched against coded dictionaries in order to identify the words. Most systems need to be 'trained' by the user with sample words, but research continues on 'speaker independent' systems that will recognise words from any speaker without training.

Work Flow System - A system which manages the flow or progress of a series of work activities or processes performed by an enterprise, department or individual. The term is also used to describe the proactive tool set used for the analysis, compression and automation of information-based business cycles.

REFERENCES

11 REFERENCES

- 1 FDA (1987)**
Guidelines on General Principles of Process Validation
- 2 OECD (1995)**
Monograph 10: The application of the principles of GLP to computerised systems
- 3 ICH (1996)**
Harmonised Tripartite Guideline for Good Clinical Practice (E6)
- 4 ICH (1997)**
Harmonised Tripartite Guideline for Statistical Principles for Clinical Trials (E9) Step 4
- 5 FDA (1997 draft)**
Guidance for Industry. Computerized Systems Used in Clinical Trials
- 6 US EPA 2185 (1995)**
Good Automated Laboratory Practices
Section 8.8, pages 2-108
- 7 FDA (1997)**
Medical Devices; Current Good Manufacturing Practice (CGMP) Final Rule;
Quality System Regulation
Federal Register October 7, 1996; 21CFR Part 820
- 8 FDA Medical Device Software Validation (1997 draft)**
Guidance for Industry: General Principles of Software Validation
- 9 FDA (1997)**
Electronic Records; Electronic Signatures; Final Rule.
Federal Register, March 20; 21CFR Part 11

BIBLIOGRAPHY

12 **BIBLIOGRAPHY**

- 1** Chamberlain, R. (1992)
Computer Systems Validation for the Pharmaceutical and Medical Device Industries
- 2** Double, M. E. and McKendry, M. (1994)
Computer Validation Compliance - A Quality Assurance Perspective
USA: Interpharm Press
- 3** FDA (1995)
Glossary of Computerized System and Software Development Terminology
- 4** FDA (1997)
Guidance for Industry: Computerized Systems Used in Clinical Trials (Draft)
- 5** Huber, L. (1995)
Validation of Computerized Analytical Systems
USA: Interpharm Press
- 6** Korteweg, M. (1993)
Computer Validation in Clinical Research: Regulatory Requirements in the European Community Drug Information Journal, 27, 315-319
- 7** PDA Committee (1995)
Technical report No.18: Validation of Computer Related Systems
PDA Journal of Pharmaceutical Science and Technology, PDA Inc.
- 8** Stokes, T., Branning, R. C. and Chapman, K. G. (1994)
Good Computer Validation Practices: Common Sense Implementation
USA: Interpharm Press

SOME USEFUL WEB SITES

13. SOME USEFUL WEB SITES

- ABPI** Association of the British Pharmaceutical Industry:
www.abpi.org.uk/
- ACDM** Association for Clinical Data Management:
www.acdm.org.uk
- ACRPI** Association for Clinical Research in the Pharmaceutical Industry:
www.dashnet.com/acrpi/
- BARQA** British Association of Research Quality Assurance:
www.barqa.com
- BIRA** The British Institute of Regulatory Affairs:
www.bira.org.uk/
- DIA** Drug Information Association:
www.diahome.org/
- EMEA** European Agency for the Evaluation of Medicinal Products:
www.eudra.org/emea.html/
- FDA** Food and Drug Administration:
www.fda.gov/
- ICH** International Conference on Harmonisation:
www.ifpma.org/ich1.html/
- MCA** Medicines Control Agency:
www.open.gov.uk/mca/mcahome.htm/
- PSI** Statisticians in the Pharmaceutical Industry:
www.psiweb.org

APPENDICES

APPENDIX I

REGULATORY GUIDELINE REFERENCES TO VALIDATION

This appendix contains extracts from relevant regulatory guidelines, which indicate the need for validation of computer systems in clinical research.

1. ICH E6: Good Clinical Practice

This guideline contains the following statement relating to the need for control and validation of computer systems.

‘5.5.3 When using electronic trial data handling and/or remote electronic trial data systems, the sponsor should:

- (a) Ensure and document that the electronic data processing system(s) conforms to the sponsor’s established requirements for completeness, accuracy, reliability, and consistent intended performance (i.e. VALIDATION).
- (b) Maintain SOPs for using these systems.
- (c) Ensure that the systems are designed to permit data changes in such a way that the data changes are documented and that there is no deletion of entered data (i.e. maintain an audit trail, data trail, edit trail).
- (d) Maintain a security system that prevents unauthorized access to the data.
- (e) Maintain a list of the individuals who are authorized to make data changes.
- (f) Maintain adequate back-up of the data.
- (g) Safeguard the blinding, if any (e.g. maintain the blinding during data entry and processing).’

2. ICH E9: Statistical Principles for Clinical Trials (Step 4)

This guideline contains the following statement relating to the need for control and validation of computer systems.

‘5.8 Integrity of Data and Computer Software

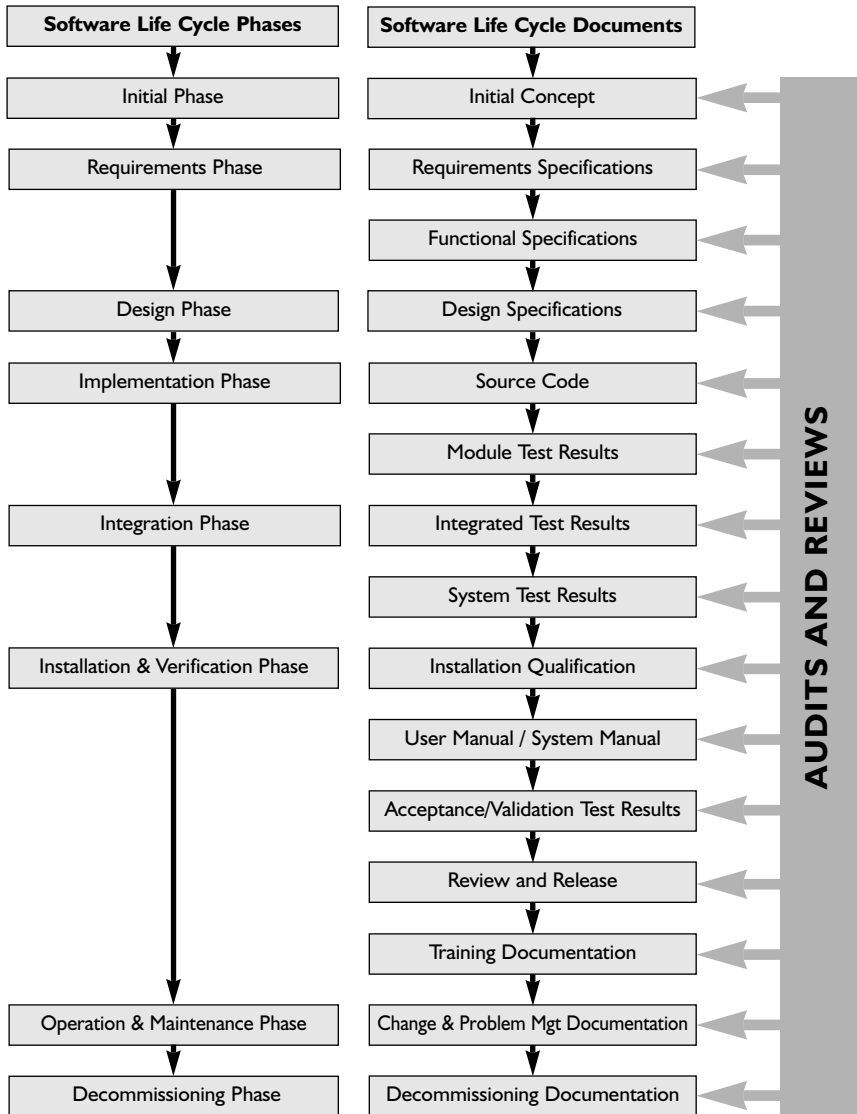
The credibility of the numerical results of the analysis depends on the quality and VALIDITY of the methods and software used both for data management (data entry, storage, verification, correction and retrieval) and also for processing the data statistically. Data management activities should therefore be based on thorough and effective standard operating procedures (SOPs). The computer software used for data management and statistical analysis should be reliable, and documentation of appropriate software testing procedures should be available.’

APPENDIX 2

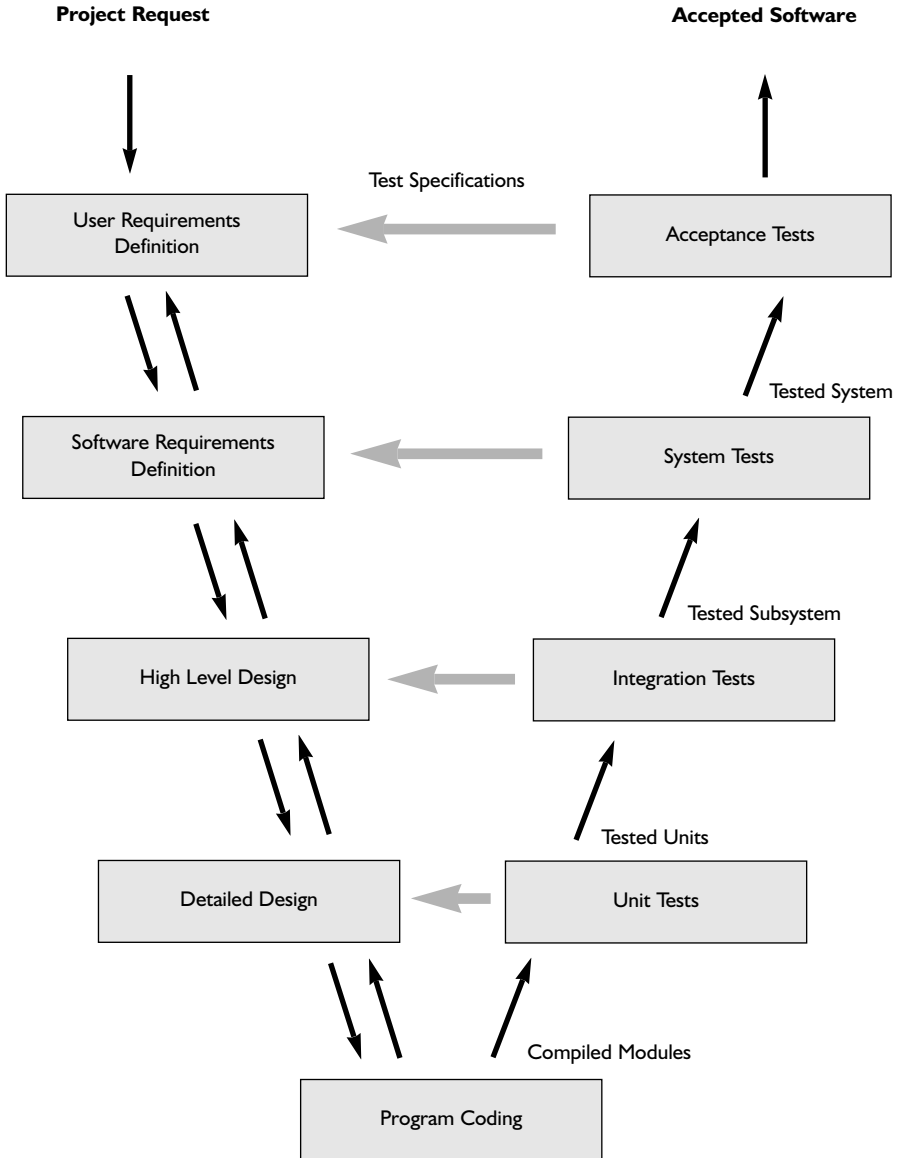
SOME COMMON SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) MODELS

- A. Classical SDLC (page 63)
- B. Classical V-Shaped SDLC (page 64)
- C. Rapid Application Development (RAD) or Prototyping SDLC (page 65)

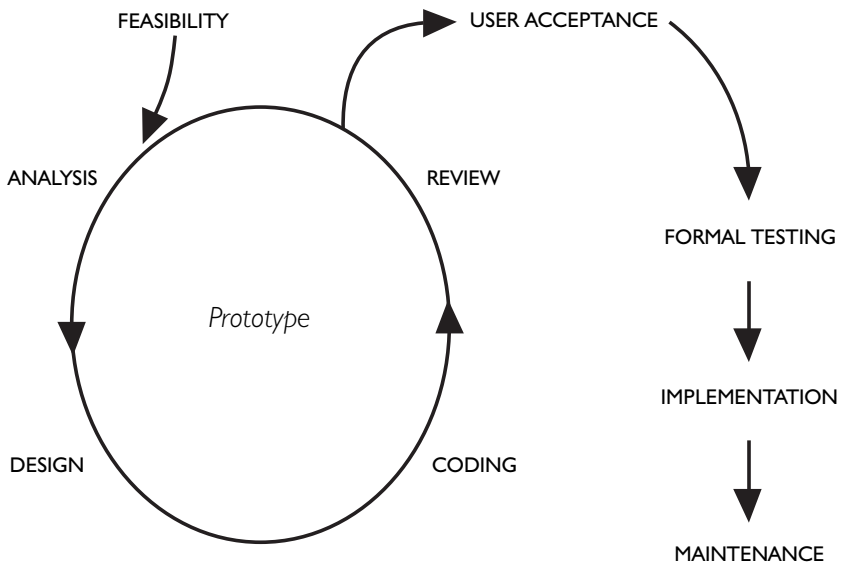
A Classical SDLC



B Classical V-Shaped SDLC



C Rapid Application Development (RAD) or Prototyping SDLC



APPENDIX 3

USER ACCEPTANCE TESTING

User acceptance testing is performed at the end of software development prior to installation. It should be performed in an environment that is as close as possible to the production environment, but separate from it. The following steps are recommended:

1. The User Requirements and Functional Specification should be referenced to consider each function of the software. The consequences of each component failing should be considered and graded, thereby directing the testing efforts.
2. A Test Plan should be devised to include all the functions of the software, performance testing (speed and response) and load testing (large but realistic volumes of input, output, processing and storage). It may also consider an assessment of the user-friendliness of the system. Justification should be given for any function not tested. At minimum, the Plan should identify the purpose of each test, required input data, keystrokes, expected outputs, documentation to be kept and required signatures. It should also allow each test to be accurately repeated and record the procedure to be followed for anomalies and any deviations from the Plan.
3. Consider the use of thread testing and automated testing (e.g. keystroke files, electronic comparisons). Note that the validation of automated testing software should also be considered.
4. Test data should be devised or acquired. The specification of test data should be directly related to the software being tested and the purpose of the test. The test data should include:
 - representative data
 - extreme but realistic values
 - boundary values
 - missing values
 - incorrect values.
5. Tests should be run and reviewed as specified in the Test Plan. Any deviations from the Plan should be noted.
6. All screens, logs and outputs should be checked for errors, warnings or anomalies and for accuracy against the expected output. All anomalies should be documented, the action taken should be recorded and the actual results obtained should be documented and retained.
7. If the test is performed according to the Plan and gives the expected outputs, the test can be signed off by the tester and reviewer.
8. If the test does not go to plan, then the anomaly procedure should be invoked.
9. The results of all the tests should be summarised with a conclusion as to acceptability.

APPENDIX 4

SOP CONTENTS

- 1. Documentation Management**
 - Security
 - Standards
 - Review and Approval
 - Distribution
 - Access Control
 - Version Control
 - Change History
 - Archiving
 - Retention Policy

- 2. Software Development and Testing Including One-off Programs**
 - Requirements Specification (Needs Analysis)
 - Functional Specification
 - Design Specification
 - Test Plan/Results
 - Testing Failures/Exception Handling
 - Change/Version Control
 - User Manual and Training (development)
 - Programming Standards (*see Appendix 8*)
 - Documentation Sign-off and Retention
 - Release and Installation

- 3. System Set-up and Installation**
 - System configuration
 - Set-up options
 - Installation tests and their documentation

- 4. User Acceptance Testing**
 - Testing methodology
 - Testing failures
 - Documentation development and retention
 - Reporting and sign-off
 - Acceptance criteria for authorisation of a valid system for use
 - Formal acceptance of the system
 - Ongoing validation

5. Training

- Training requirements for staff involved in the validation of computer systems
- Training records for computerised systems
(This may be part of the general SOP on training and records.)

6. Security

- Password assignment and changes
- Access authority levels and their review
- Physical security
- Virus checking and prevention

7. System Use and Maintenance

- An SOP for each system (reference User Manual)

8. User Support

- Support/help for problems
- User Manuals and training (maintenance)

9. Problem Management

- Problem log
- Monitoring and resolution

10. System Back-up and Restoration

- Routine backups
- Backup log
- Restoration of system and data

11. Business Continuity

- System recovery in the event of a major disaster

12. Change Management

- Documentation and maintenance of the system configuration
- Documentation, prioritisation and authorisation of changes to validated systems (hardware, software, procedures and people)
- Development, testing, documentation and review of changes
- Installation of changes and provision of training

13. Periodic Review

- Revalidation requirements
- Re-testing

14. Decommissioning

- Authorisation
- Data conversion to new systems
- Decommissioning Plan

15. Archiving

- Validation documentation
- Obsolete Software
- Maintenance of old systems

16. Audit

- In-house software development including one-off programs
- Vendor Audit
- Implementation documentation
- In-house maintenance of the validated state
- CRO computer system (pre-study inspection and during study)
- CRO one-off programs

APPENDIX 5

DOCUMENTATION CHECKLIST

The following list suggests documents for inclusion in the validation document set of a project, or for inspection during an audit. Not every document listed will be required for every system validation or audit, nor for every re-validation of the same system. While common core documents are listed, it is recommended that the validation document set requirement be assessed for each project individually, since each project may incur special risks, necessitating additional documentation.

Note that although there is substantial overlap between documents in the list and routinely produced project management documents, not all project management documentation will necessarily be appropriate to validation, nor vice versa.

All documents should be version controlled and signed to indicate ownership and ensure traceability, e.g. by author, approver, authoriser.

Category	Document	Source
QA	SOPs (see Appendix 4) Systems Analysis Standards Programming Standards	QA Systems Analysis Development
Validation	Validation policy Validation plan Validation report	Corporate Management Validation Project Manager Validation Project Manager
System Design	Requirements specification Hardware and operating environment specification Database design specification Technical specification Installation instructions	Systems Analysis Systems Analysis Systems Analysis/ Database Administration Development Development
Impact Assessment	Impact on SOPs, working practices and guidelines Impact on other computer systems Security implications	User Development, Owners of other systems Development, Database Administration, User/Systems Management
Test Strategy	System test strategy User acceptance test strategy	Development User
Test	Source code review Module test plans Modules test results System test plans System test results Database integrity test plans User acceptance test plans User acceptance test reports	Development Development Development Development Development Database Administration User User

Category	Document	Source
Problem Tracking & Management	Pre-release incident reports	Development/User /Database Administration
	Pre-release incident management	Development
	Post-production release incident reports	User
	Post-production release incident management	Development
Process Control	Authorisation for release to test environment	User
	Ratification of validation report	Senior Management, User
	Authorisation for release to production	User
	Release notes	Development
Personnel	CV	All involved staff
	Job Description	All involved staff
	Record of training & experience	All involved staff
	Competency records	All involved staff
Training	User guide	Development, User
	Training materials	Training
	User training records	Training
System Maintenance	Service level agreement for system maintenance	Development, Vendor
Change Management	Register of changes to hardware and software	Validation Project Manager
Business Continuity	Back-up log	System Management

NOTE: (a) The Validation Project Manager should be trained in validation, represent the users and be independent of the development team.

APPENDIX 6

POTENTIAL CAUSES OF VALIDATION FAILURE

Potential causes of validation failure include the following:

- Inadequate documentation of plans.
- Inadequate definition of what constitutes the computer system.
- Inadequate definition of the expected results.
- Inadequate specification of the software (e.g. user requirements, functional specification).
- Software does not meet its specification.
- The source code for the software is not available.
- Inadequate specification of the computer hardware and operating environment for which the system is designed to work.
- The computer hardware or operating environment differs from the specification.
- The way the system should be used is not defined.
- Inadequate consideration given to the centralised IT infrastructure, e.g. network management, procedures and responsibilities.
- The intended use of the system is clearly defined, but users are not aware of it, or do not adhere to it.
- The system has been inadequately tested, or the testing has been inadequately documented.
- Documented standard procedures for the development, maintenance, operation (including security) or use of the system are inadequate.
- Documented procedures for disaster recovery are inadequate.
- System developers or other personnel involved with system implementation and use are not properly qualified, trained or competent.
- Documentary evidence to demonstrate qualification, training or competence level of personnel involved with the system is not available.
- Documentation for all or part of the validation process does not exist, or cannot be located.
- Evidence of review and approval of validation documentation by qualified staff is not available.
- Inadequate change control over any element of the system (i.e. hardware, software, procedures, people).

APPENDIX 7

VALIDATION PLAN CONTENTS

The following summarises the broad areas that a Validation Plan should address:

1. Document Control

A document control sheet indicating the current version of the plan together with the revision history of the document.

2. Approval of Plan

A sheet indicating the signatories for approval of the Validation Plan.

3. Introduction and Objective

A brief statement as to what system is being developed and the purpose of the document.

4. Description of System

A brief description of the system, its use, and the hardware, operating system and network software on which it will run.

5. Risk Assessment

A list of the risks to the validated state of the system and the extent to which they are addressed by the Plan.

6. Scope

A description of the extent of the validation exercise including:

- constraints on the validation activities
- assumptions made for the successful execution of the Validation Plan
- boundaries to the validation activities, i.e. a description of the points beyond which the validation testing will not run
- exclusions of any specific areas within the scope with reasons why.

7. Tasks and Responsibilities

A checklist of tasks and who is responsible for their execution, review and approval.

8. Validation Maintenance

A statement as to how the validated system will be maintained thereafter. This may refer to other procedures in place and should cover change control, security and access, business continuity, training and periodic review.

9. Referenced Documents

A list of documents, procedures and other supporting documentation referred to throughout the Validation Plan.

APPENDIX 8

SAMPLE PROGRAMMING STANDARDS

Software should be designed and programmed according to source code development standards to ensure consistency and quality. Examples of commonly used programming practices are described below.

I. Naming Conventions

I.1 Files

File names should be descriptive and reflect the functions or contents of the files. They should contain only alphanumeric characters (possibly plus the underscore character), and should always start with a letter, not a number.

I.2 Directories

It is recommended that files are stored in a protocol-specific directory structure. However, if it is more practical to place the file in a drug-specific directory, then the protocol number(s) should be incorporated into the file naming convention.

I.3 File Extensions

For operating systems that support file name extensions, a standard file extension naming convention should be used, e.g.

filename.SAS - SAS program

filename.SQL - SQL program

filename.DAT - ASCII data file

filename.LOG - SAS log file

filename.TXT - ASCII text file

I.4 Directory Index

It is recommended that an index file is created (e.g. INDEX.TXT) for each directory that contains a list of all the programs/files in that directory with a short description of the contents/function of each file.

I.5 Variables

Variable names should be meaningful and reflect the contents of the variable. If it is difficult to select a meaningful name, it is helpful to assign a descriptive label to it where possible.

2. Program Documentation

All programs and subroutines should contain documentation that precedes the source code in the form of a header comment block. The following information should be included:

Program Name	Name of the program
Platform	VAX, Windows, DOS, UNIX, etc.
Version	Version of the software (e.g. 6.12 of the SAS package)
Author(s)	Name(s) of the programmer(s)
Date	Program Creation Date
Drug/Protocol	The drug name and/or protocol number of the study
Purpose	A description of what the program does and why it exists
Parameters	Description of variables received by or passed back from the program.
Data Files	List any data sources for the program (e.g. ASCII files, ORACLE tables, permanent SAS data sets, etc.)
Programs Called	List any other programs called which are external to the program.
Output	List any output files generated by the program.
Assumptions	List any assumptions upon which the program relies.
Restrictions	List any restrictions of the program.
Invocation	Describe how the program is executed.
Modification History	This contains information for change control, including the date of each modification, the name of the programmer making the modification, and the reason for and description of each modification. This section should be updated each time a new modification is made.

Documentation should appear throughout the program to give the reader a good overview of the function and structure of all areas of the program. There should be at least one comment for each main step, new idea, or algorithm within the program. When a step or algorithm is complex, further comments should be added alongside the lines of code. Note that commenting should complement the code, and not hinder readability.

3. Program Layout

- Each source code statement should appear on a separate line.
- A blank line should be left between each logical section in the source code to aid readability.
- Indenting should also be used to aid the readability of the code. Statements should be indented to show the hierarchical relationship that exists between them. For example, for IF/THEN/ELSE statements, DO/END statements, etc., the body of statements contained in the IF/THEN or DO/END block should be indented at least 2 characters to the right with respect to the IF or DO

statement, i.e.

if (*condition*) then do;

statement;

statement;

end;

else do;

statement;

statement;

end;

- d) All variables should be declared and initialised at the beginning of the program. Default data types should not be used.
- e) All non-executable statements (e.g. variable declarations) should be grouped together preferably at the beginning of the program.
- f) Complex mathematical expressions should be simplified by separating terms with spaces, or by breaking down the complex expression into a number of simple expressions.

4. General Practices

- a) It is good practice to arrange code into small re-usable modules. Once such modules have been validated, always try to re-use this validated code as much as possible.
- b) Possible program input and execution errors should be predicted in advance and handled appropriately in the source code (e.g. division by zero).
- c) Avoidance of undesirable practices is also important to ensure the program does not process the data in unexpected ways under unexpected conditions. Examples of practices to avoid include:
 - 1. commented out code in final versions of programs
 - 2. hard-coded data changes in non-conversion programs
 - 3. data processing that varies with patient or visit number.

5. Output Labelling

Output should be labelled with:

- a) the identity of the source program, including version number
- b) the date and time generated
- c) the identity of the user
- d) the page number and total number of pages.

APPENDIX 9

GUIDELINES FOR TESTING ONE-OFF PROGRAMS

Ideally, testing should be performed both during and upon completion of program development. The recommended steps involved in the testing process for one-off programs are as follows:

1. A Test Plan should be devised to exercise each major component of the program. At minimum, the plan should specify the purpose of each test and required input data with associated expected outputs, including any relevant boundary values.
2. Relevant test input data sets should be devised or acquired. This data should contain extreme, but realistic, values in order to adequately stress the software. [NOTE: This is not needed for a one-off program where there is a single known data set]

The test data could come from various sources, such as:

- Random Numbers, e.g. for verifying a modelling algorithm.
- Records from a live database such as Oracle, e.g. for verifying the internal data structures of a program which were created with a knowledge of the database structures.
- Static, hard-coded, or program-generated data, e.g. for testing a simple algorithm for which a finite number of possible input values can exist.

The choice of test data should be directly related to the program(s) being tested, and the purpose of the test.

3. Run the tests specified in the test plan against the test input data set(s).
4. Any program log files, etc. should be reviewed for the presence of warning and/or error messages. The presence of notes in the log file could also suggest a problem.
5. It should be confirmed that any data sets created by the program(s), including intermediate data sets, contain the expected number of observations.
6. The actual output should be compared with the expected output. This can be done in a number of ways, including:
 - Check the output against that produced by an already validated program which gives similar output.
 - Take a random sample of patients from each treatment group, and manually check the data listings for these patients against the records in the master database.
 - Check the calculated values for derived variables for a sample of patients by hand-calculations from raw data.
 - For statistical analyses, check that the appropriate data records have been used in the analysis (to ensure that withdrawals, protocol violators,, etc. have been handled appropriately), check the degrees of freedom are as expected, check for outliers, and check the underlying assumptions of the analysis.

The method(s) used to check the output, together with the results of this checking, should be documented. This documentation should provide enough information to allow the checks to be duplicated.

7. If all expectations are met and there are no discernible errors or anomalies, then the testing process is complete. Otherwise, the programmer should correct the errors or anomalies and repeat steps (3) to (7) of the testing process until the program yields the expected output.
8. The test output should be signed and dated. Associated print-outs and documentation pertaining to the tests, including the Test Plan, should be archived with the clinical study file.

APPENDIX 10

PUBLISHED STANDARDS AND GUIDELINES FOR SYSTEM/SOFTWARE DEVELOPMENT AND VALIDATION

1. British Standards Institute (BSI)

- BSI 6719 (1986) Guide to Specifying User Requirements for a Computer Based System.
 BSI 7799 (1995) Code of Practice for Information Security Management.

2. Federal Information Processing Standards (FIPS)

- FIPS 101 (1983) Guideline for Lifecycle Validation, Verification and Testing of Computer Software.
 FIPS 105 (1984) Guideline for Software Documentation Management.
 FIPS 132 (1987) Guideline for Software Verification and Validation Plans

3. International Standards Organisation (ISO)

- ISO 9001 (1994) Quality Systems - Model for Quality Assurance in Design/Development, Production, Installation and Servicing.
 ISO 9000-3 (1990) Guidelines for the Application of ISO 9001 to the Development, Supply and Maintenance of Software.
 ISO/IEC 9126 (1991) Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their Use.
 ISO/IEC 12207 (1995) Information Technology - Software Life Cycle Processes.

4. Institute of Electrical and Electronic Engineers (IEEE)

- IEEE 610.12 (1990) Glossary of Software Engineering Terminology.
 IEEE 730 (1989) Standard for Software Quality Assurance Plans.
 IEEE 828 (1990) Standard for Software Configuration Management Plans.
 IEEE 829 (1993) Standard for Software Test Documentation.
 IEEE 830 (1990) Recommended Practice for Software Requirements Specification.
 IEEE 983 (1986) Guide to Software Quality Assurance Planning.
 IEEE 1012 (1992) Standards for Software Verification and Validation Plans.
 IEEE 1016 (1987) Recommended Practice for Software Design Descriptions.
 IEEE 1016.1 (1993) Guide to Software Design Descriptions.
 IEEE 1028 (1996) Standard for Software Review and Audits.
 IEEE 1042 (1987) Guide to Software Configuration Management.
 IEEE 1059 (1993) Standards for Verification and Validation Plans.
 IEEE 1063 (1987) Standard for Software User Documentation.
 IEEE 1074 (1991) Standard for Developing Software Life Cycle Processes.
 IEEE 1219 (1992) Standard for Software Maintenance.



INDEX

A

Acceptance Testing, 4, 14, 15, 19, 21, 24, 57, 66, 67
Access, 18, 24, 27, 30, 35, 39, 41, 43, 45, 49, 50, 51, 52, 54, 55, 56, 57, 61, 66, 68, 74
Access Control, 27, 35, 39, 49, 67
Access Rights, 21
Approval, 7, 11, 17, 19, 23, 29, 35, 49, 50, 51, 55, 67, 73, 74
Archived data, 18, 31
Archiving, 19, 28, 45, 49, 51, 55, 67, 69
ASCII files, 76
Audit Report, 37
Audit Trail, 40, 44, 45, 49, 52, 55, 61
Audits, 31, 35, 36, 37, 38, 80, 63
Authorisation, 29, 34, 51, 52, 67, 68, 69, 72
Auto-encoding, 44, 46

B

Bar Coding Systems, 42
Benefits, 11
Business Continuity, 19, 24, 27, 28, 68, 74, 72

C

Calibration, 36, 38, 40, 42, 49
Change Control, 23, 24, 34, 37, 55, 73, 74
Change Management, 19, 27, 28, 29, 36, 30, 68, 72
Changes, 11, 18, 19, 20, 21, 25, 27, 28, 29, 31, 36, 40, 41, 45, 49, 54, 55, 61, 68, 72, 77
Clinical Database Management Systems, 43
Clinical Research Computer Systems, 7, 9
Coding Dictionaries, 43, 45
Commissioned Systems, 18, 21, 24, 35, 47
Compliance, 11, 18, 35, 36, 37, 47, 55, 56, 59
Configuration Control, 44
Configuration Management, 41, 80
Consistency Checking, 44
Contract, 12, 35, 36, 37, 43
Contract Research Organisations, 12, 35, 43
Control, 11, 12, 18, 19, 23, 24, 27, 28, 29, 31, 34, 35, 36, 37, 39, 41, 42, 46, 49, 50, 55, 56, 57, 60, 61, 67, 73, 74, 76, 72
Corporate Management, 7, 17, 71

D

Data Capture, 7, 39, 41, 44, 52, 54, 55
Data Capture Systems, 39, 44, 54
Data Checking, 40, 44
Data Corruption, 27, 43, 51
Data Entry, 38, 40, 41, 44, 46, 55, 57, 61
Data Entry Systems, 40
Data Entry Verification, 40, 50, 57
Data Identifiers, 44
Data Loading, 44, 45, 46
Data Tables, 45
Data Transfer, 40
Data Validation, 44
Data Verification, 40, 50, 57
Database, 7, 39, 40, 41, 43, 44, 45, 46, 48, 50, 52, 53, 55, 56, 78, 71, 72
Database Locking, 45
Database Snapshot, 45
Database Unlocking, 45
Date and Time Stamping, 45
Decommissioning, 7, 9, 11, 17, 18, 19, 27, 31, 69, 63
Derived Data, 43, 44, 46, 47, 55
Developer, 12, 28, 33, 34, 47, 15, 73
Development, 7, 9, 11, 12, 17, 19, 21, 23, 24, 29, 33, 34, 35, 37, 45, 46, 48, 51, 53, 55, 56, 57, 59, 62, 65, 66, 67, 68, 69, 73, 75, 78, 80, 65, 71, 72
Dictionary, 43, 44, 46, 55
Disaster Recovery, 28, 43, 73
Documentation, 7, 14, 15, 18, 19, 21, 22, 23, 24, 25, 28, 29, 30, 33, 34, 35, 36, 37, 38, 40, 41, 45, 50, 55, 56, 57, 61, 63, 66, 67, 68, 69, 70, 73, 74, 76, 78, 79, 80
Dummy Data, 43, 45
Duplicate Records, 44

E

Edit Checking, 44
Electronic Diary Cards, 41
Emergency Changes, 29
Entry Screens, 43
Environment, 12, 18, 21, 24, 27, 28, 29, 31, 37, 39, 41, 45, 47, 48, 55, 56, 57, 66, 73, 71, 72
Environmental Changes, 31

Escrow Agreements, 22
 Evaluation Plan, 25, 56
 Evaluation Report, 25, 56
 Extraction of Data, 45

F

Format, 19, 25, 40, 43, 44, 45, 49, 56

G

GCP, 11, 18, 35, 55

H

Help Desk, 41

I

ICH GCP, 35
 Implementation, 7, 18, 19, 21, 28, 29, 46, 57, 59, 69, 73, 63, 65
 Intelligent Character Recognition, 42, 56
 Internet, 40, 43, 56
 Investigational Site Audits, 37
 Investigator, 11, 12, 35, 37, 38, 40, 41, 43, 53

L

Laboratory Data, 44, 46
 Legacy Systems, 18, 21, 25
 Libraries, 43, 44
 Logical Security, 27, 38
 Loss, 27, 28

M

Macros, 45, 47
 Maintaining the validated state, 7, 27, 14, 15
 Maintenance Level, 36, 38, 63
 Measuring Devices, 39, 40
 Misuse, 28

O

Operating Environment, 18, 24, 27, 37, 47, 48, 73
 Optical Character Recognition, 42, 56

P

Password, 27, 41, 43, 50, 52, 56, 68
 Personnel, 12, 15, 17, 18, 23, 24, 27, 28, 33, 36, 40, 41, 55, 57, 73, 72
 Physical Security, 27, 68
 Plausibility Checking, 44, 50
 Policy, 7, 12, 14, 15, 16, 17, 18, 51, 67, 71
 Problem Management, 19, 28, 68

Production Version, 29, 31
 Program Development, 45, 78
 Programs, 12, 14, 15, 18, 19, 29, 33, 34, 36, 40, 45, 47, 53, 57, 67, 69, 75, 76, 77, 78

Q

Quality Assurance, 18, 31, 35, 56, 59, 60, 80
 Quality Control, 18, 28, 31, 35, 56
 Quality Management, 11, 14, 15, 16, 18, 35, 56
 Quality System, 11, 35, 36, 37, 38, 39, 56, 58

R

Range Checks, 43
 Rapid Application Development, 17, 21, 56, 60, 65
 Reaching the validated state, 21, 14, 15, 22
 Reference data, 43, 45
 Reference Ranges, 43
 Regulations, 19, 20, 36
 Regulatory, 7, 9, 11, 12, 14, 15, 35, 39, 43, 48, 53, 54, 55, 56, 59, 60, 61
 Remote Systems, 40
 Replacement System, 28
 Report, 12, 22, 23, 25, 34, 37, 40, 48, 53, 56, 57, 59, 71, 72
 Reporting, 9, 24, 28, 45, 46, 48, 55, 67
 Reports, 44, 45, 48, 51, 53, 54, 55, 71, 72
 Requirements, 11, 12, 18, 28, 33, 36, 39, 40, 41, 43, 47, 48, 49, 52, 53, 55, 56, 57, 59, 61, 63, 66, 67, 68, 73, 80, 63, 64, 71
 Responsibilities, 17, 18, 19, 22, 23, 27, 33, 57, 73, 74
 Restoration, 19, 27, 39, 41, 46, 68
 Restore, 28
 Retrieval, 19, 45, 46, 49, 50, 53, 55, 56, 61
 Retrospective Evaluation, 21, 25, 57
 Revalidate, 28
 Review, 3, 7, 19, 23, 27, 31, 34, 35, 45, 46, 48, 49, 51, 53, 55, 67, 68, 73, 74, 80, 63, 65, 71
 Risk, 11, 12, 17, 21, 23, 31, 33, 74
 Risk Assessment, 11, 31, 74

S

Scope, 11, 12, 17, 19, 52, 57, 74
 Security, 19, 27, 36, 38, 39, 40, 41, 43, 49, 50, 51, 52, 57, 61, 67, 68, 71, 73, 74, 80
 Service Agreements, 36, 38
 Software Development Life Cycle, 17, 24, 48, 57, 62
 Source Code, 24, 33, 34, 55, 56, 57, 63, 71, 73, 75, 76, 77
 Specifications, 9, 21, 33, 43, 56, 57, 63, 64
 Standard Data Checks, 44
 Standard Operating Procedures, 18, 61
 Standard programs, 14, 15, 18, 33, 47

Support, 19, 24, 28, 41, 56, 57, 68, 75
 System Error, 11, 28
 System use and maintenance, 19, 28, 68
 Systems Developed by Third Parties, 16, 23
 Systems Developed Internally, 35

T

Templates Programs, 45
 Templates, 43, 50, 53
 Test Data, 31, 47, 66, 78
 Test Plan, 31, 34, 37, 57, 66, 67, 71, 78, 79
 Testing, 7, 11, 12, 14, 15, 19, 21, 23, 24, 25, 29, 31, 34, 37, 39, 40, 42, 46, 49, 51, 55, 56, 57, 61, 65, 66, 67, 68, 73, 74, 78, 79, 80
 Time Frozen Representation, 45
 Training, 7, 18, 19, 23, 24, 27, 28, 30, 35, 37, 38, 41, 42, 48, 49, 57, 63, 67, 68, 72, 73, 74, 75

U

Unauthorised Access, 27
 Upgrades, 50
 User Acceptance Testing, 14, 15, 19, 21, 24, 57, 66, 65, 67
 User Control, 24
 User Management, 7, 17, 27, 29, 57
 User Support, 19, 24, 28, 68

V

Validation, 3, 7, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 25, 28, 30, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 61, 63, 66, 67, 68, 69, 70, 71, 72, 73, 74, 80
 Validation document set, 23, 70
 Validation Failure, 14, 15, 21, 73
 Validation Plan, 14, 15, 21, 22, 23, 25, 57, 71, 74, 80
 Validation Process, 7, 11, 16, 17, 21, 35, 73
 Validation report, 22, 23, 34, 57, 71, 72
 Validation Status, 31, 39, 57
 Validation, commissioned systems, 18, 24, 47
 Validation, in-house systems, 12, 14, 18, 23, 47
 Validation, legacy systems, 14, 18, 21, 25
 Validation, third party systems, 14, 18, 23, 24, 27, 37
 Vendor audit, 37, 40, 41, 42, 69
 Vendor Control, 23, 24, 37
 Vendors, 23, 35, 40
 Version Control, 18, 19, 29, 36, 46, 49, 50, 57, 67, 70
 Viruses, 27



ACDM
PO Box 129
Macclesfield
Cheshire
SK11 8FG
Tel. +44 (0) 1625 511818
Fax +44 (0) 1625 511750

PSI
PO Box 126
Macclesfield
Cheshire
SK11 8FH
Tel/Fax +44 (0) 1625 511750