ELSEVIER

Journal of Pharmaceutical and Biomedical Analysis
14 (1995) 13-22

JOURNAL OF
PHARMACEUTICAL
AND BIOMEDICAL
ANALYSIS

Practical computer validation for pharmaceutical laboratories [☆]

R.D. McDowall [1]

Department of Chemistry, University of Surrey, Guildford, Surrey, GU2 5XH, UK

Received for review 25 April 1995; revised manuscript received 24 July 1995

Abstract

Computer validation should address the scientific correctness of the application software, the business objectives of the organisation, and the concerns of regulatory agencies. It is a balance between a practical and cost effective system that must develop the confidence that the system is under control. This paper gives an overview of computer validation from the following perspectives.

What is computer validation?

Why can a computer system not be completely validated?

Why and how should computer systems be validated?

The principles outlined in this paper should be adapted to a specific system on a case by case basis depending on its scope and complexity.

Keywords: Computer validation; Verification; System development life cycle; Revalidation policy

1. Computer systems and pharmaceutical industry regulations

During the past twenty years, good manufacturing practice (GMP) and good laboratory practice (GLP) regulations have been promulgated for pharmaceutical manufacturing or research and development. These regulations were aimed at ensuring the production of quality analytical results and final product. The emphasis, during the early day, was the ensure that manual procedures were documented and followed so that at any later time the analysis could be checked.

Computerisation of laboratories has increased since the introduction of these regulations which have struggled to keep up progress in this area. Computerisation comes in many shapes and sizes; these can be classfied as follows:

- Commercial software which is purchased from a third party and used without any modification in the laboratory, such as the basic functions of a spreadsheet;
- Commercial software which is purchased from a third party and then customised to the intended operation. This can vary from a procedure within a spreadsheet through a chromatographic data system to a laboratory information management system (LIMS).

☆ Presented at the Sixth International Symposium on Pharmaceutical and Biomedical Analysis, April 1995, St. Louis, MO, USA.

[1] Address for correspondence: 73 Murray Avenue, Bromley, Kent BR1 3DJ, UK. Tel. & Fax: +44-181-313-0934; E-mail: 100411.1142@Compuserve.com.

SSDI 0731-7085(95)01629-5

- Bespoke or purpose built software, where the function of the application is built from software tools such as a database or programming language. This can range from small programs written by an individual to a large-user system such as a LIMS.
- Embedded software within laboratory instrumentation for operations such as isocratic or gradient HPLC pump functions and chromatography integrators.
- Programmable logic controllers (PLCs) which are custom programmed, usually in-house, to undertake specific tasks, often around pharameceutical manufacturing operations.

Therefore, computerised systems are pervasive within the laboratory. The speed of systems development has usually outstripped the formulation of regulations. This has lead to one of two approaches:

(i) The issue of addenda to a country's regulations with specific guidance for computer systems. Typical examples are the Japanese [1] and UK [2] GLP guides and European Union GMP Annex 11 [3].

(ii) To equate computers with equipment and apply the existing guidelines. As such computerised systems should be fit for purpose, have adequate capacity and have the same data integrity, accuracy and security as manual procedures. This is the general approach of the US FDA and the principles of GLP as applicable to computer systems have been outlined by Lepore [4].

In both GLP and GMP areas, industry groups have worked, sometimes with active regulatory agency input, to produce more detailed documents for implementation guidance for computer validation. Such publications have come from the Drug Information Association (DIA) [5], Pharmaceutical Manufacturer's Association (PMA) [6] and the United Kingdom Computer Validation Forum (UKCVF) [7].

2. Why validate computerised systems?

There are a number of reasons for validating computerised systems. First, to ensure consistent product quality. Product quality can be used in the widest scope: the product of a laboratory is information, therefore from the research and development laboratories validation is ensuring that the information from all laboratories is correct and timely and the data contained within the reports are auditable. Computerised systems are highly involved with manufacturing the final product and it is important to know that these are functioning correctly to ensure consistent quality of final product.

Second, compliance with regulations, both the FDA and EU [3,4] expect manual and computerised systems to show equal quality. Good validation practices will ease or expedite regulatory inspections and audits and reduce the risk of non-compliance. Third, confidence in computerised data enables a good foundation for management control especially throughout a multinational company which can be evidenced with better communication across teams and also with regulators.

Last, but not least, computerised systems validation affords investment protection. The investment in computerised systems has risen dramatically over the past decade, but what is the success rate? The apocyryphal statistic in the LIMS arena is that 50% of systems fail to meet initial expectations. Validation is a way of building quality into a product and increase the odds that the system will meet expectations.

The risks associated with poor or no validation can be summarised as delays in submission to regulatory authorities, product recalls, negative publicity associated with Form 483 citations made public under the Freedom of Information Act, or even the shutdown of the manufacturing plant. The citations associated with computer validation can be grouped into six categories [8]:

- Evidence of testing;
- Evidence of training;
- Evidence of audit and review;
- Evidence of management responsibility;
- Evidence of design control;
- Evidence of document control.

Validation must address all of these issues, not only during the development of a system, but also during its operational life.

The cost of regulatory non-compliance to a pharmaceutical laboratory can be an increased number and/or repetition of studies in drug development, an increased time for approval, shorter product life on the market, a poor reputation with regulatory agencies and possibly difficulty attracting the best scientists.

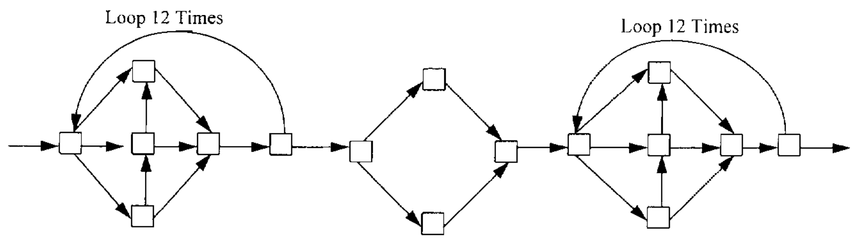

Fig. 1. The complexity of computer validation. This simple program flow segment has 10^{20} different pathways through it (Boehm 1970).

3. Computer validation: the problems

The validation of a computer system has a number of problems associated with it.

3.1. Self regulation

Regulatory agencies take the view that the end-users of a system are responsible for its validation. The agencies will audit the system and will inform you if there are any problems with the work you have done. This is not very satisfactory as the end-users can rarely perform more than black box testing unless they have detailed knowledge of the design specification of the system and the aid of skilled computer scientists.

3.2. What am I to do?

This leads to the problem of how to interpret the guidelines in a cost effective approach to validation. Often many iterations of trial and error can be involved, where validation is either over-engineered or not sufficiently rigorous.

3.3. Complete validation of a system is a myth

Unless there is a very simple system, it cannot be completely validated. This was demonstrated eloquently by the work of Boehm [9] who described the simple program flow segment shown in Fig. 1. The number of pathways, and hence possible tests of the software, in this segment was calculated to be 10^{20}. If one makes an absurd assumption that one test per second can be conceived, designed, executed and documented then it will take more than three times the geological age of the earth to validate this program segment. Unfortunately most computer systems are far more complex.

3.4. Different regulations and terminology

Pharmaceutical companies may face a number of different regulations or guidelines for computer validation. Given the progress made to date through the International Conferences on Harmonisation (ICH), perhaps computer validation may be a suitable candidate for harmonisation in the future. Agreement on the scope and content of regulations affecting computer systems would be welcome.

The use of GMP and GLP within different sections of the industry has led to the situation where different terminology can be used by the two groups, typically around the terminology for installation qualification (IQ), operational qualification (OQ), and performance qualification (PQ) for GMP which have the equivalent of validation plan and validation report for GLP.

3.5. Consistency of inspectors

The human element, in the form of what will pass without comment with one inspector but not another, will never completely disappear. The computer literacy of inspectors is increasing and with this will come increased scrutiny of computerised systems, far more so than now. However, consistency of regulatory approach and inspection is highly desirable.

Notwithstanding these problems, this paper, will look at the scope of computer validation and see what steps the prudent laboratory or organisation can implement to ensure the adequacy of their systems.

4. The scope of computer validation

Validation is defined as *established documented evidence which provides a high degree of assurance that a specific process will consistently produce a product meeting its predetermined specification and quality attributes* [10], and as a prerequisite of regulatory guidelines.

The key concepts in this definition are:
- documented evidence;
- high degree of assurance;
- consistency and reproducibility;
- predetermined specification and quality.

Note that in this definition there is no mention of computerised systems: it is applicable to all processes.

In general, validation is concerned with generating the proof to demonstrate that the computerised system is accurate when validated and continues to be so when it is operational and that there is sufficient evidence of management control. This usually means that an action must be documented. Another feature of validation is to produce an auditable system; having the appropriate documentation to aid inspection thereof.

The problem posed by validation is shown in Fig. 2. On the left is the stimulus to computer validation posed by regulations or guidelines. Any

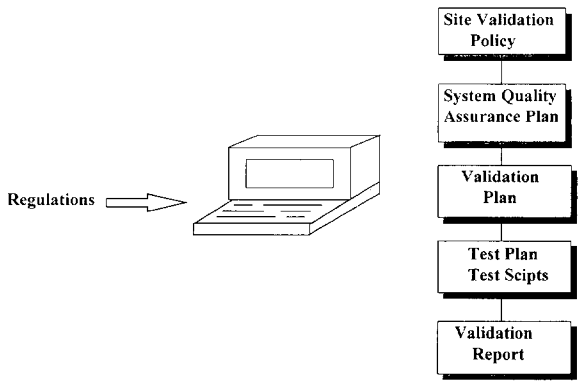

Fig. 2. Initial validation of a computer system in response to a regulatory stimulus.

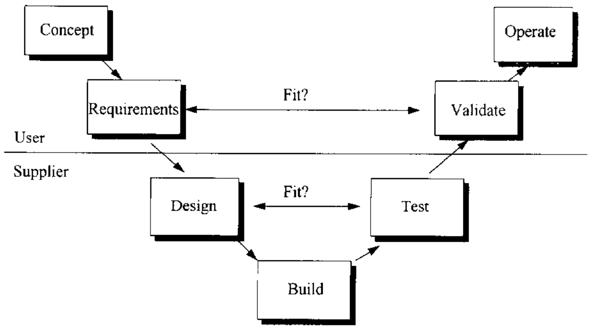

Fig. 3. System development life cycle model.

response to this stimulus should be:
- logical;
- scientifically lucid;
- structured;
- reflect the way you use the application;
- be in the environment you work in.

This latter point is most important, *there is no point validating a function of a system that is not used.*

Computer validation must provide confidence to first and foremost laboratory management and users, secondly to an internal quality audit and thirdly to external inspector. Inspectors only audit the laboratory on a periodic basis. All others work in the laboratory and use its computerised systems daily. The users must have the confidence in a system above all others, otherwise the investment will be wasted.

5. Life cycle validation

The life cycle approach [11,12] should design and build a quality system and should, ideally be covered by a corporate validation policy. From this, an overall system quality assurance plan for each individual system, a validation plan, test plan and a validation report should be written. These aspects are covered in more detail in the papers by Stokes [13], Segalstad and Synnevad [14] and Moore et al. [15].

The life cycle is shown in Fig. 3, and goes from the concept of the system, through the user requirement specification, design specification, building and coding, testing and validation before the system becomes operational. Once operational there can be modifications over the life time of the system which must be validated and authorised before they become operational. Finally, the system is retired when it is obsolete and/or no longer in every day use, and its replacement is commissioned.

The life cycle is shown in the shape of a "V". The left hand side of the "V" is the design of the system, at the bottom is the building stage and the right hand side is the testing phase. Looking horizontally across the "V", the validation of the system is concerned with testing against the user requirements. Therefore it is imperative that a user requirements specification exists to base the validation on. *If there is no user requirement specification document there can be no validation of the system.* This area is the responsibility of the end-users of the system and their management.

5.1. *White box and black box testing*

Validation by the end-user should be based upon the way that the system is used in a particular laboratory. Therefore, a computer application cannot be considered validated by one laboratory simply because another laboratory has validated the same software. The end-users will not be able to undertake technical testing either because they do not have the full technical specification of the system or they do not possess the technical skills to undertake this type of testing. Black box testing requires only the knowledge of the purpose of the module and the end results [16]. Therefore, users will undertake black box testing, where known inputs will be entered and the outputs compared with that expected.

In contrast, white box testing requires the full knowledge of the complete specification and full observability of the inputs and outputs and processing within each module of the application. This is usually only available to system developers and programmers and is rarely afforded to end-users. The design specification is used to devise tests to prove that the functions described work as designed.

6. A methodology for validation

A public domain methodology exists to structure the validation efforts, furthermore, it is mandated by the United States Government for use by Federal Departments. This is the Manual of Software Engineering Standards [17], devised the published by the Institute of Electronic and Electrical Engineers (IEEE). This book is revised on a regular basis where new standards are added and existing ones are either revised or reaffirmed regularly. A number of standards are directly relevant to computer validation including: 729.1983, Glossary of Software Engineering Terminology; 730.1–1989, Software Quality Assurance Plans (SQAP); 829–1983, Software Test Documentation; 1012–1986, Software Verification and Validation Plans.

Two standards (829–1983 and 1012–1986) were used a the basis of the GLP approach to validation for computers used in non-clinical studies [5]. Stokes [13] has recently published a paper on the use of these standards in computer validation and an overview of their use is presented here.

6.1. *Validation and verification*

These two concepts are very important and are defined by the IEEE [18] as:

- Validation is the process of evaluating software at the end of the software development process to ensure compliance with software requirements. Validation answers the question "is it the right product for a specified application?".
- Verification is the process of determining whether or not the products of a given phase of the software development life cycle fulfill the requirements established during the previous phase. Verification answers the question "is the product built right?".

Validation takes place at the end of the life cycle; it is the responsiblilty of the end-user. The intention of the testing effort is for success (including those tests which are deliberately designed to fail). As there is usually a time lag between the purchase of the system and when validation takes place, if major programming errors or design problems are found now, much time will be spent rectifying the retesting the changes. Validation is much more than merely demonstrating the correctness of a computerised system: it covers the computer environment and includes training and the documented operation of the system.

Verification, in contrast, takes place at the end of each stage of the life cycle, before the next stage in the process begins; it is more rigorous and is aimed at ensuring the quality of the requirements, design and building stages. Verification is designed to ensure that the user requirements are actually incorporated into the final application. Verification is a very important process that is usually overlooked in many computerised system projects which usually leads to inability of that system to meet initial expectations and/or rejection by the users. Verification aims to find errors. The more quality time spent on the design and its verification; the better product will result and more robust its application.

7. Validation and verification in practice

The approach to validation, based on the use of the IEEE standards is shown in Fig. 4. The key documents are the validation plan, the test plan for an individual phase of the life cycle, the test scripts for the executed life cycle phase and a test summary report generated for that phase of the life cycle. When the whole life cycle is complete a

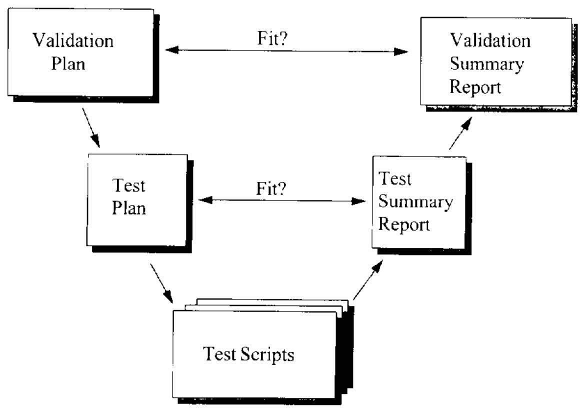

Fig. 4. Validation model based on the IEEE approach to validation.

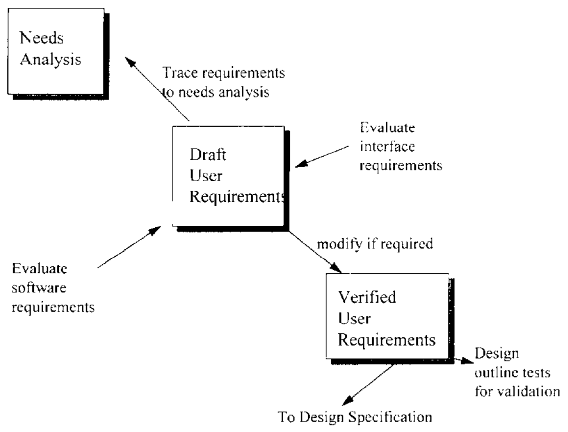
Needs Analysis
Trace requirements to needs analysis
Draft User Requirements
Evaluate interface requirements
Evaluate software requirements
modify if required
Verified User Requirements
Design outline tests for validation
To Design Specification

Fig. 5. Verification of the user requirements specification.

validation summary report is written. Note the emphasis on summary reporting for validation activities. These IEEE standards can be used per se or modified as appropriate, indeed the latter is the case with the DIA document [5].

The approach to validation is shown in the shape of a "V". Like the life cycle this represents the design of validation on the left of the "V", execution of validation at the base and summary reporting on the right of the "V". Similar to the life cycle in Fig. 3, there are checks for fit from the validation plan to the validation summary report and the test plan and test summary report. The key sections of the main parts of the document are presented by Stokes [13].

Verification is useful at the end of each stage of the system development life cycle, e.g. the writing of the user requirements specification which is the description of what the users want the system to do and is usually derived from a needs analysis. The verification of the draft user requirements specification is shown in Fig. 5. One has to trace the user requirements back to the needs analysis document and verify the completeness, traceability, accuracy and consistency [19] for each requirement. Ideally, an independent group of users should also evaluate the same document. Any interfacing requirements should also be verified; interfaces between analytical instruments and other computer applications are very important and should be checked as thoroughly as possible. If any missing requirements of inconsistencies have been found, the user requirements specification must be amended and then approved by management [19]. The verified user requirements specification then goes to the next stage of the life cycle; the design phase.

The extra work in ensuring that the user requirement specification is correct in time and resources well spent. Problems that should have been rectified at this stage are far more expensive to solve further into the life cycle. When the user requirements specification of complete, the outline validation tests can be generated.

The tests should target the functions of the system that will attract the most regulatory concern such as data capture, integrity, manipulation, and capture. Consider an analogy, and full scope of a computerised system is represented by a target (Fig. 6). This is what the user *could* validate if they had sufficient time and resource, however this is impossible from the work of Boehm. The bull's-eye contains the functions that *must* be validated as they attract the most regulatory attention. The inner represents the system functions that *should* be validated if there was the time and resources available. Therefore a validation plan should be constructed around the way the system is used in the laboratory and concentrate on the *must validate* functions which should be linked by as many of the *should validate* functions as is practicable.

The test scripts [13] are the heart of any validation effort and will take the most time to write (Fig. 7). The concept is that the test script will form the testing log and the archive for the actual

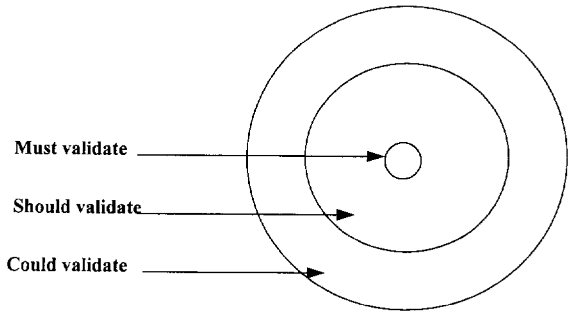

Fig. 6. Targeting validation: the system functions can be defined as must validate, should validate and could validate.

testing. All experimental data and output will be recorded here.

Some of the types of testing that could be carried out are as follows.

- Boundary test: the entry of valid data within the known range of a field, e.g. a pH value could have acceptable values within 0–14.
- Stress test: entering data outside of designed limits, e.g. pH value of 15.
- Predicted output: knowing the function of the module to the tested, a known input should have a predicted output.
- Consistent operation: important tests of major functions should have repetition built into them to demonstrate that the operation of the system is reproducible.
- Common problems: both on the operational and support aspects of the computer system should be part of any validation plan. The predictability of the system under these tests should generate confidence in its operation.

The test summary report [13], summarises the testing effort and if anyone wishes to see the actual work they can be shown the completed test scripts. The validation summary report [13], brings together all of the testing throughout the whole of the life cycle and presents a recommendation for management approval when the system is validated. The IEEE methodology' uses summary reporting as a rapid and efficient means of presenting results as the detail is contained in the test scripts.

7.1. Validation roles and reponsibilities

According to Reugge [16] there are four groups that can be involved in validation of computerised systems: the users, management, the quality assurance unit and the vendor. A fifth group should also be considered: the information sciences or computer sciences group that may be involved in the support of larger or networked systems. The roles and responsibilities of each group will be discussed below.

The role and responsibilities of the users are to define the functions of the system, verify its installation and to define and execute the validation plan. Users will need to have standard operating

Test Script Identifier Unique identifier for the test script
Purpose Describe briefy the system or software feature to be tested
Special Requirements Any special requirements that are necessary for this procedure e.g. specific skills of the tester or linkage of this test with another test script
Test Procedure Steps 1. Describe the procedure that the tester should carry out 2. Write the procedure exactly as the tester will carry out the test 3. Cross reference to any documents such as a user manual
Test Log Describe the expected results Write in the observed results found during testing
Unexpected Events Describe any unexpected events such as the test not working as anticipated or errors occuring during the test. If no unexpected events have occured, state 'none'
Resolution of Unexpected Events Describe any steps to resolve any unexpected events. This section will not be applicable if no unexpected events have happened
Pass / Fail Criteria State what the pass and fail criteria are for the test. Compare the expected and observed results: are they the same? Does the test pass or fail?
Sign Off by Tester and Peer Reviewer

Fig. 7. Outline test script for the validation or verification of a computerised system.

procedures written for operating and supporting the application, the user base must be trained and users must ensure that the complete documentation of the system is available for audit and inspection. Although the end-user is responsible for these areas, they need help, advice and support in this.

Active support by management is essential for making the resources available for the validation effort and to take the responsibility for authorising the use of the system in the regulated environment. Furthermore, management must encourage the participation of the Quality Assurance Unit (QAU) in this process.

The QAU provides assistance in the interpretation of regulatory guidelines for comuterised systems and reviews the documentation produced during the validation effort. Monitoring of the testing and validation effort and offering assistance in developing SOPs, are additional roles and responsibilities for the QAU. If there are any vendor audits to be undertaken, then the QAU should be involved in the planning and execution of this activity. However, some QAU personnel may not be very computer literate, but this must change as many regulations involving computerised systems require the active involvement of the QAU [1,2]. In fact, the Japanese Regulatory Authority stipulates that QAU should have access to the system via their own terminal [1].

The Information or Computer Sciences Group of the organisation may be involved, in the purchase of larger applications and may also support the system during its operational life. In offering support to a regulated area, the IS group become bound by the regulations of that area. Many larger applications such as chromatography data systems or LIMS are supported by a central group, moreover, PC software applications such as spreadsheets are served throughout an organisation by the same group. What is not often realised both by the users and the IS group is that any unauthorised change to the operating system or network will make a validated system non compliant. One way of overcoming this problem is to have a service level agreement between the user community and the IS group where the responsibilities and accountability of all groups are set out in a contract.

If a computerised system has been purchased from a vendor, then the vendor can be involved in the validation process by providing tools, documentation, services, and possibly additional resources. It also means that the whole life cycle will not be performed by the one organisation. Therefore audits of the vendor practices should be undertaken to verify that the vendors quality practices have been followed and a quality system has been developed. Similar to the approach of individual pharmaceutical organisations as to how far their laboratories will support GMP or GLP, the same is true for ISO 9000 accrediation by commercial organisations.

The fact that a company has ISO 9000 accrediation does not mean that it is the right product for a laboratory nor that it has been built correctly . Each individual laboratory or organisation should evaluate it against their own user requirements specification. Once the product has been selected the vendor should be audited to sec if the ISO 9000 quality system is operated effectively. The evaluation and audit process is a very important part of the life cycle as it ensures the design, build and testing stages (which are under the control of the vendor) have been checked to ensure compliance with the regulations. The audit should be planned and cover items such as the design and programming phases, product testing and release, documentation and support; a report of the audit should be produced after the visit.

Some vendors offer "validation" certificates for some of their products. These should be verification certificates, as the end-user is responsible for validation and this cannot be developed to a third party. However, a laboratory can take the verification work that the vendor had done and incorporate it into its testing effort?

Equally, vendors also offer help with validation by selling off the shelf validation scripts. These must be evaluated very carefully against the appropriate regulatory guidelines and the way you use the system in your laboratory. Check that the scripts match your operation and that they allow for both boundary and stress testing of the system. Otherwise these scripts are potentially dangerous as they can lead a laboratory into a false sense of security, they are convenient to purchase and the laboratory initially knows no better.

Further resources for programming additional functions, installing the system or training may be services offered to a laboratory by a vendor.

8. Maintaining validation during operational use

After a system is validated and it becomes operational, changes will usually occur during its operational lifetime which may impact the validation status. The issues that a laboratory may face are:

- revalidation criteria;
- configuration management;
- change control;
- audit trails;
- standard operating procedures (SOPs);
- operational logs and maintenance records;
- error logging and resolution.

Each is an essential component of validation and can be applied, in all or part, to any item of laboratory computer equipment be it an integrator, a central chromatography data system, a robot or a LIMS. All that is required as a modification of this approach, for example if an instrument does not have the means to back up data, there is no need for a log to record this activity. The aim of this coverage is to provide the confidence that the laboratory's management is in control of the system. These topics have been discussed in greater detail in a recent publication [20] and the reader is referred to this paper for more information.

9. Conclusions

Validation of a computerised system is more than just testing at the end of the system development life cycle. It is concerned with designing quality into the product from the start of a project. Validation at the end of the life cycle must be complimented by verification at each stage, this is a more cost effective approach. Testing should be targeted at those system functions that attract the most regulatory concern.

References

[1] Koseisho, Good Laboratory Practice Attachment: GLP Inspection of Computer System, Pharmaceutical Affairs Bureau, Ministry of Health and Welfare, Tokyo, pp. 157–160, 1988.
[2] Department of Health, Good Laboratory Practice, United Kingdom Compliance Programme: The Application of GLP Principles to Computer Systems, Advisory Leaflet Number 1, Department of Health, London, 1989.
[3] Good Manufacturing Practice for Medicinal Porducts in the European Community, Annex 11, Commission of the European Communities, Brussels, 1992.
[4] P.D. Lepore, Chemometrics Intell. Lab. Syst.: Lab. Inf. Manage., 17 (1992) 283–286.
[5] Computerized data systems for nonclinical safety assessment — Current concepts and quality assurance, Drug Information Association, Maple Glen, PA, USA, 1988.
[6] Computer Systems Validation Committee of the Pharmaceutical Manufacturing Association, Pharm. Technol., 10(5) (1986) 24–34.
[7] Good Automated Manufacturing Practice, Pharmaceutical industry supplier guidance for validation of computer related systems in pharmaceutical manufacture, United Kingdom Computer Validation Forum, Second Draft, October 1994.
[8] S. Weinberg, in M.D. Hinton (Ed.), Laboratory Information Management Systems: Development and implementation for a quality assurance laboratory, M. Dekker, New York, 1994.
[9] B. Boehm, Some information processing implications of air force missions 1970–1980, The Rand Corporation, Santa Monica, 1970. Quoted by S.S. Herrick, Validation of computer systems, 4th Annual Quality Assurance Roundtable, Houston, Texas, 28 September 1983.
[10] General Principles of Validation, Food and Drug Administration, Centre for Drug Evaluation and Research, Rockville, MD, USA, May 1987.
[11] R.D. McDowall, Chemometrics Intell. Lab. Syst.: Lab. Inf. Manage., 13 (1991) 121–133.
[12] S. Segalstad, Chemometrics Intell. Lab. Syst.: Lab. Automation Inf. Manage., in press.
[13] T. Stokes, Chemometrics Intell. Lab. Syst.: Lab. Automation Inf. Manage., in press.
[14] S. Segalstad and M.J. Synnevad, Chemometrics Intell. Lab. Syst.: Lab. Inf. Manage., 26 (1994) 1–12.
[15] J. Moore, P. Solanki and R.D. McDowall, Chemometrics Intell. Lab. Syst.: Lab. Automation Inf. Manage., in press.
[16] D. Reugge, presented at 4th International Symposium on Automation, Robotics and Artificial Intelligence applied to Analytical Chemistry and Clinical Laboratory Medicine, Montreux, February 1995.
[17] Software Engineering Standards Collection, Institute of Electronic and Electrical Engineers, Piscataway, NJ, USA.
[18] IEEE Standard 729-1983, Glossary of software engineering terms, Institute of Electronic and Electrical Engineers, Piscataway, NJ, USA.
[19] IEEE Standard 1012-1986 Software validation and verification plans, Institute of Electronic and Electrical Engineers, Piscataway, NJ, USA.
[20] R.D. McDowall, Chemometrics Intell. Lab. Syst.: Lab. Automation Inf. Manage., in press.