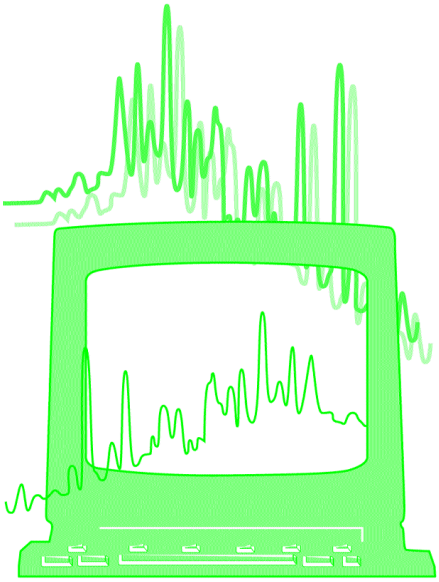

Chromatography Data Systems IV: Managing Change in a Changing World

R.D. McDowall, McDowall Consulting, Bromley, Kent, UK.

How do you maintain the validation status of your chromatography data system?

Imagine the situation, you are sitting at your terminal late on a Thursday afternoon trying to get enough work finished to make Friday an easy day, when disaster strikes. A message appears on the screen that the computer systems will be unavailable over the weekend as the latest version of the network operating system will be installed.

Did you know about this upgrade? No, of course not! The IT department never tells you about things like this, but it doesn't make any difference to you or your validated chromatography data system (CDS) does it? Of course not!? Unfortunately it does.

When you return on Monday and try to log on, assuming that the upgrade goes without any problems, your previously validated CDS is no longer validated. A change has taken place that is

- unplanned
- uncontrolled
- untested (apart from when you log on and try to use it)
- undocumented.

Oh dear, what can we do?

As I wrote in the last article in this series (1), the initial validation effort for a CDS from concept to becoming operational is the easy part of the project. This may take anywhere between 4 and 12 months, depending on the size of the project. However, this is just the start of the validation voyage. You will have made a large investment in time and effort to validate the system. The lifetime of a data system can be anything up to 10 years, and you'll have to maintain the validation status of the system throughout that time if you want to maintain the quality of the data you're generating. The question is will you sink or swim during the validation voyage?

Let's consider the challenges that we all face when dealing with maintaining the validation of a CDS or indeed any system. The main theme of this article is managing change in a changing world. Let's look at some of the types of changes that will impact an operational CDS:

- finding software bugs and installing associated fixes
- upgrades to application software, operating systems, plus any software tools or middleware used by the CDS
- network improvements, such as changes in hardware, cabling, routers and switches to cope with increased traffic and volume
- hardware changes, such as PC and server upgrades or increases in memory, disk storage etc.
- interface to new applications (e.g., spreadsheets or laboratory information management systems (LIMS))
- expansion or contraction of the system because of work or organization reasons
- environmental changes: moving or renovating laboratories.

All of these changes need to be controlled to maintain the validation status of your CDS.

In addition, there are also other factors that impact the system from a validation perspective, such as:

- problem reporting and resolution
- software errors and maintenance
- back-up and recovery of data
- archive and restore of data
- maintenance of hardware
- disaster recovery (business continuity planning)
- written procedures for all of the above.

We'll be looking at a number of measures that need to be in place to maintain the validation status of a CDS operating in a regulated or accredited laboratory. The principles outlined in this article should be adapted to the CDS on a case-by-case basis: for instance, an integrator with no means of storing data should not require a back-up log in contrast to a client-server system that does.

Regulations and Guidelines

Here's the boring bit — you can skip to the next section if you want. Just as the regulations and guidelines provide a view on what is expected during the implementation and release of a CDS, they provide views on what they expect to see during the operational phase of the system. In general, the emphasis is concerned with generating the proof to demonstrate that the computerized system is accurate when validated and continues to be so when it is operational, and that there is sufficient proof of management awareness and control. To obtain proof of an action usually means that it must be documented, although the format of documentation (paper, magnetic or optical) is left open by all schemes.

Let's have a look at what the regulations say. The key sections of European Union (EU) Good Manufacturing Practice (GMP) Annex 11 for computerized systems are shown in Table 1 (2). Please note that this is a selective presentation of the clauses that are applicable for the operational use of an application; for a full picture I suggest that you read the whole of the regulations.

The key point of each clause of EU GMP in Table 1 is summarized below with a cross-reference to the clause:

- validation covers the whole lifecycle (11.2)
- environmental conditions must be within specifications (11.3)

- system description (11.4)
- access control and user account management (11.8)
- audit trails for data quality (11.10)
- change control procedures (11.11)
- data back-up quality and security (11.13)
- data back-up (11.14)
- alternative ways of working (11.15)
- procedures for breakdown (11.16)
- problem identification and resolution (11.17).

Organization for Economic Cooperation and Development (OECD) regulations (3) have similar approaches to maintaining the validation status of operational systems; however, these have included the minimum requirements for written procedures. These should cover, but not be limited to, the following:

- procedures for the operation and use of computerized systems (hardware/software), and the responsibilities of personnel involved
- procedures for security measures used to detect and prevent unauthorized access and program changes
- procedures and authorization for program changes and the recording of changes
- procedures and authorization for changes to equipment (hardware/software) including testing before use if appropriate
- procedures for the periodic testing for correct functioning of the complete system or its component parts and the recording of these tests
- procedures for the maintenance of computerized systems and any associated equipment
- procedures for software development and acceptance testing, and the recording of all acceptance testing
- back-up procedures for all stored data and contingency plans in the event of a breakdown
- procedures for the archiving and retrieval of all documents, software and computer data
- procedures for the monitoring and auditing of computerized systems.

It is important to realize that if you are working to GMP complementary information from Good Laboratory Practice (GLP) regulations is available and vice versa.

T1

Table 1: Main Clauses from EU Annex 11 Applicable for Maintaining the Validation Status of an Operational CDS.

11.2 The extent of validation necessary will depend on a number of factors including the use to which the system is to be put, whether the validation is to be prospective or retrospective and whether or not novel elements are incorporated. Validation should be considered as part of the complete life cycle of a computer system. This cycle includes the stages of planning, specification, programming, testing, commissioning, documentation, operation, monitoring and modifying.

11.3 Attention should be paid to the siting of equipment in suitable conditions where extraneous factors cannot interfere with the system.

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

11.8 Data should only be entered or amended by persons authorized to do so. Suitable methods of deterring unauthorized entry of data include the use of keys, pass cards, personal codes and restricted access to computer terminals. There should be a defined procedure for the issue, cancellation, and alteration of authorization to enter and amend data, including the changing of personal passwords. Consideration should be given to systems allowing for recording of attempts to access by unauthorized persons.

11.10 The system should record the identity of operators entering or confirming critical data. Authority to amend entered data should be restricted to nominated persons. Any alteration to an entry of critical data should be authorized and recorded with the reason for the change. Consideration should be given to building into the system the creation of a complete record of all entries and amendments (an "audit trail").

11.11 Alterations to a system or to a computer program should only be made in accordance with a defined procedure which should include provision for validating, checking, approving and implementing the change. Such an alteration should only be implemented with the agreement of the person responsible for the part of the system concerned and the alteration should be recorded. Every significant modification should be validated.

11.13 Data should be secured by physical or electronic means against wilful or accidental damage, in accordance with item 4.9 of the guide. Stored data should be checked for accessibility, durability and accuracy. If changes are proposed to the computer equipment or its programs, the above mentioned checks should be performed at a frequency appropriate for the storage medium being used.

11.14 Data should be protected by backing-up at regular intervals. Back-up data should be stored as long as necessary at a separate and secure location.

11.15 There should be available adequate alternative arrangements for systems that need to be operated in the event of a breakdown. The time required to bring the alternative arrangements into use should be related to the possible urgency of the need to use them. For example, information required to effect a recall must be available at short notice.

11.16 The procedures to be followed if the system fails or breaks down should be defined and validated. Any failures and remedial action taken should be recorded.

11.17 A procedure should be established to record and analyse errors and to enable corrective action to be taken.

Change Control and Configuration Management

When I perform audits of any operational computer system, the first place that I start my investigation is to look at the changes to the system over any period of time throughout its operation. The reason is that most computer systems change over time for a variety of reasons as we discussed at the start of this article. Changes always occur even to an integrator that uses firmware. As few systems remain in their initial configuration for long, it is essential to track all modifications to a system over time. Again this reiterates the original purpose of many quality guidelines: being able to repeat conditions under which the work was originally done.

The key question that needs answering from an inspector's perspective is whether there is demonstrable control of these changes. In many instances there is no control of the changes and, therefore, the system is out of control.

Definition of terms: Definition time! There are a number of terms we need to consider here; the first two are

- Change control: The systematic process by which any change to a computerized system is proposed, coordinated, evaluated, rejected or approved, and implemented (including testing and revalidation as necessary).
- Configuration management: The system for identifying the configuration of

hardware, software and firmware at discrete points in time with the purpose of systematically controlling changes to the configuration and maintaining the integrity and traceability of the configuration throughout the system life cycle.

These two terms are very closely linked and some organizations have condensed them to "change management" to cover all aspects of the control of a CDS, or indeed any computerized system. Note also that configuration management can also be applied to software development and refers to the control of the versions of the software modules produced. However, for the purposes of our discussion we will use it only in the wider context of the configuration of the CDS as we'll see below.

Going further in defining terms for configuration management, we have

- Configuration item: Definition of the individual components in a configuration management system. Items can include hardware (server, PC), software (application, software utilities, operating system) and peripherals (A/D units, printers). It is very important that each configuration item is carefully defined: if too detailed the process will be too resource intensive to operate but if set too high the information generated will be useless.
- Configuration baseline: The establishment of the initial configuration of the computerized system from the configuration items. If a system undergoes rapid expansion, it may be necessary to redefine the baseline.

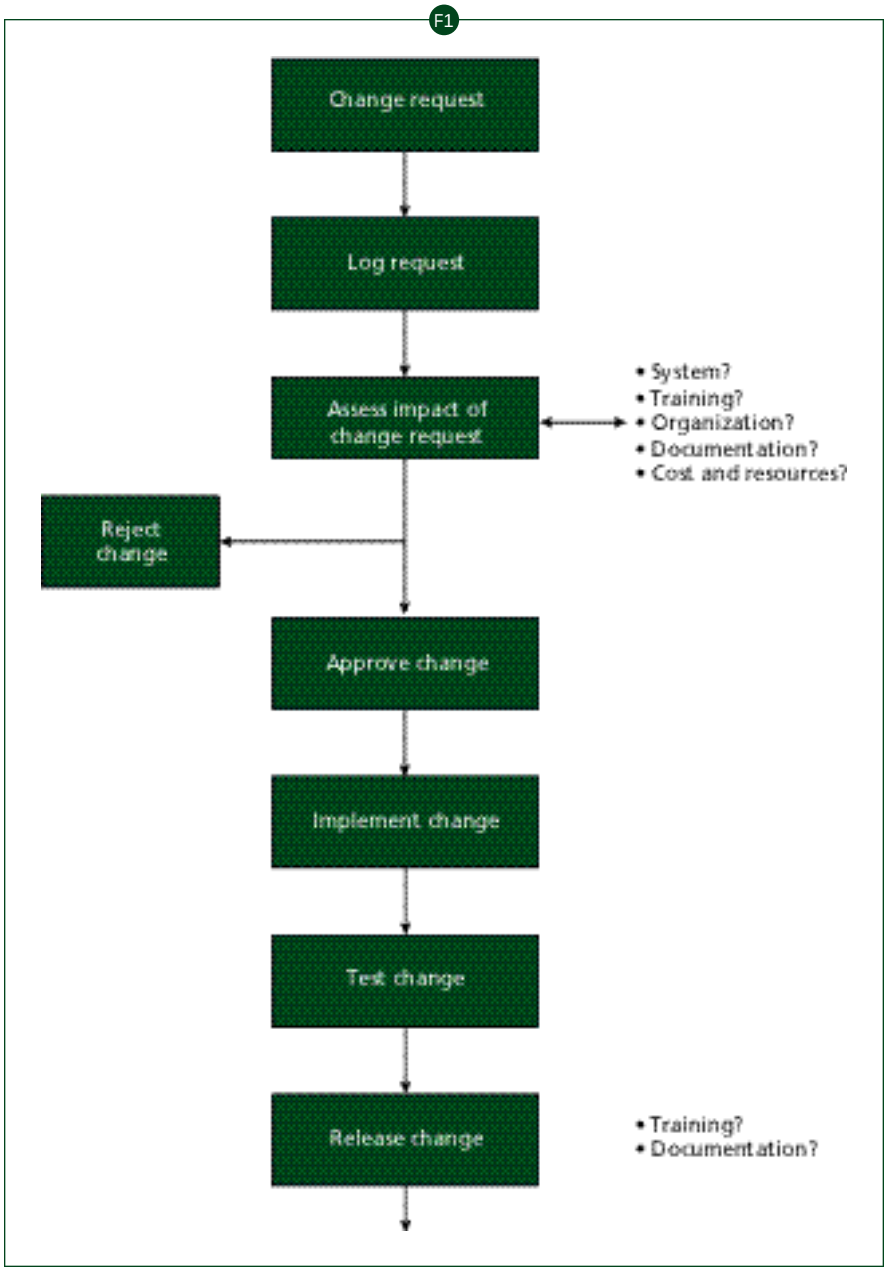

Figure 1: Change-control scheme.

Change control: According to Nakagawa (4), an established change-control process is important. Although the book the information comes from is about LIMS, the principles for change control are the same for a CDS. The change-control process should aim to establish an environment conducive to open discussions and exchange of views. The stakeholders in the system can be asked for their views, ideas and possible solutions: in short, any input to improve the quality and performance of the system. This approach is intended to avoid the situation outlined at the start of this article in which unannounced changes can destroy the validation status of all systems running on a network. Hence the importance of a coordinated approach to managing change.

A typical change-control system is characterized by the following criteria:

- responsibilities of all parties are defined
- managed process
- documented process.

The process is outlined in Figure 1 and is based, in part, on the work of Nakagawa (4) and Chapman (5) and, in part, on my personal experience.

Roles and responsibilities: The three roles outlined in the last article in the series (1) are still important in a regulated or accredited laboratory for change control and configuration management, as well as all other aspects of validation.

- User: overall responsibility for the operation of the CDS under the regulations and guidelines, especially for deciding whether to implement changes to the system. Also responsible for maintaining the configuration records of the system.
- IT manager: subcontracted by the users to provide technical expertise to assess the technical feasibility and impact of changes. The IT manager does not implement changes to the operating environment or the system unless authorized and operates the subcontracted functions of the system in a compliant way.
- Quality assurance: provides compliance information and advice and reviews key documentation to assess compliance with regulations. Performs periodic reviews of the system to ensure conformance with the regulations and guidelines.

See the section on Validation roles and responsibilities in the previous article for other responsibilities in validation for these three roles (1).

Change-control process: The first part of the process is a request for change; this requires basic information such as:

- who requested the change?
- nature of the change
- justification for the requested change.

The request for change may result from a variety of reasons. First, it may be the reporting of a bug or feature of the CDS software that should be resolved, the performance of the system is not normal or there is a request for additional resources such as a printer, A/D unit, PC or extra disk space. Whatever the change, it needs to be documented. The way for doing this should be as simple as possible, keeping the paperwork to a minimum and encouraging all that use the system to comply with the process. Alternatives that may be used in larger organizations are the use of a central help desk that can undertake the documentation of reported requests or the use of electronic mail with standard change request forms.

Second, the request needs to be analysed for its impact. There are a number of facets to consider here: the effect of the change, for example, its impact on the laboratory, the organization and also on the system itself. In looking at the impact of the change on the laboratory, one should consider the following:

- time required to implement the change
- cost of the change (including documentation and training)
- resources required (both physical and human) to make the change
- benefits of making the change.

When looking at the impact of the change on the system, consider the following points:

- Does the change provide a major or minor business benefit?
- Is the change cosmetic only?
- Is there any impact on the system?
- Are the functions already available?
- If the change is implemented will it cause any problems? (e.g., training, documentation, etc.)
- How much retesting and revalidation is required?

What is the effect of the change on the organization?

- This has a large continuum: from no impact through to greater effort required to use the system after the change has been made.
- Does the change bring a cost saving to the organization or is more cost required?
- Will the change allow for time or cost savings?
- What impact will the change have on the documentation of the system?
- What impact will the change have on the users of the system: will there be any necessity for retraining?
- What is the impact and cost of doing nothing?

Once the impact analysis has been completed, each change can be reviewed. A management group involving the major stakeholders in the system (users, IT and Quality Assurance) can undertake this process. Alternatively, this can be devolved to a small validation or change-control team consisting of two or so individuals authorized to consider and recommend changes. The size of the CDS, the business benefit and the magnitude of the change should decide the approach.

Here changes will be reviewed and can be classified into those that bring major or minor benefits. The prioritization of authorized changes will probably need to be balanced with the available budget and resources, as it is unlikely that all authorized changes will proceed. There will inevitably be change requests that will be rejected for a variety of reasons. Regardless of the decision by the reviewing group, it is of supreme importance that decisions and the rationale for making them are fed back to the requester.

Third, if the change is rejected the submitter will be informed of the rejection and the reason for it. However, if the request is approved, the resources will be made available to implement the change. The first stage is to formulate a plan to implement the change. This will incorporate any relevant aspects of the impact assessment and any technical issues such as the extent of retesting and revalidation of the CDS, update of documentation and retraining of users etc.

The change is then made and the system released for use. Hold on a minute! Have we forgotten anything here? Are you proposing to make changes to a live and operational CDS? Think again. You should consider a test environment that is at least logically (i.e., on the same computer system), or ideally a physically separate environment for making and testing the changes and then rolling out the tested changes to the production environment. Remember also that validation must occur, at least in part, on the operational system, so ensure that everything is fully backed up before you start. But you had already thought of that hadn't you?

Changes to a CDS: Figure 2 is a stylized view of a client/server chromatography data system with a client and a server linked via a network. Both the server and client consist of hardware, the operating system and the CDS application software. Note this is a stylized representation and may not represent all data systems. The diagram is reproduced courtesy of Tilo Schumacher of Pfizer, Germany.

I would like to use this diagram as a means of discussing any changes to the CDS to illustrate their impact. Consider the following possible changes to any CDS and the impact that each would have.

- changes to the network such as replacement of the cabling or upgrade of hubs or routers. Will these have any impact on the operation of the CDS?
- changing a PC client from a 133 MHz to 500 MHz PC

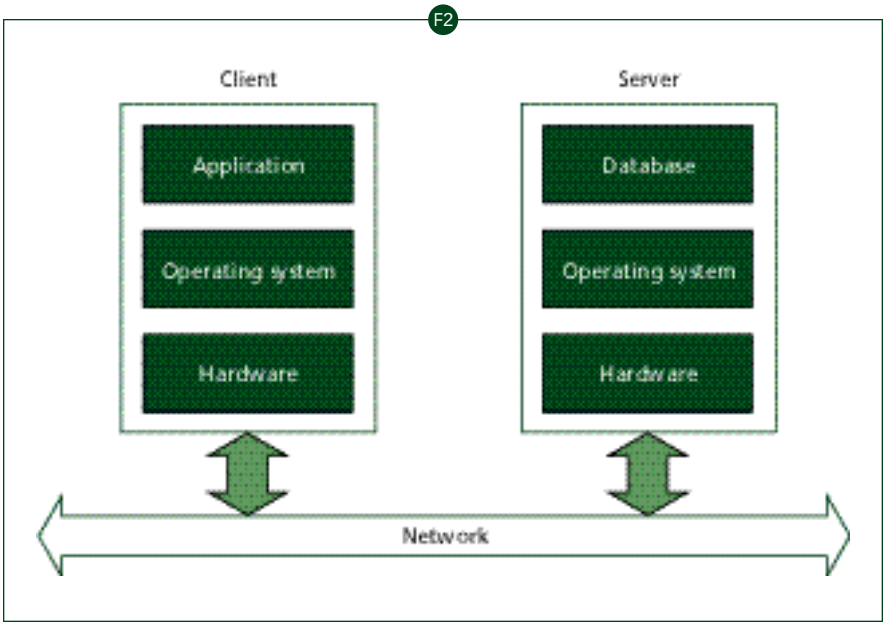

Figure 2: Stylized representation of a chromatography data system. (Courtesy Tilo Schumacher.)

- updating the operating system from Windows NT, version 4, service pack 3 to service pack 5
- fitting a software patch to the CDS software to fix a software error
- installing a new version of CDS software.

The impact that each potential change could have on the validation status must be assessed. For example, the hardware change from a 133 to 500 MHz client PC is relatively small compared with the upgrade from NT, version 4, SP3 to SP5 or the new version of CDS software.

Configuration management: Configuration management, as we defined above, is a set of procedures to ensure adequate identification, control, visibility and security of any changes made to

- hardware
- firmware
- network
- software, including any patches and macros
- specialized equipment associated with the application (e.g., the A/D units for our CDS)
- peripherals (e.g., printers and plotters).

Furthermore, all modifications should be authorized before a change is made and the personnel making the changes should also be authorized to do so by management via the change control process as outlined above. Therefore, configuration management and change control are very closely linked.

The aim of configuration management is to demonstrate that the system is under control and all modifications to it are tested and validated where appropriate. From the information in the configuration management log, either the current or any previous versions of the system can be recreated. Recreation can be done at any time; it aims to safeguard the laboratory and the user against loss of data and also enables the user to see the impact of a problem if one is found after a change has been implemented.

The process of configuration management is quite simple, first the initial configuration is established and then all changes are tested, authorized and monitored.

- Establish the baseline (initial) configuration. This is the compilation of a list comprising all the components of the system: all the release numbers and serial numbers (where appropriate) of the application software program(s), the software tools (e.g., database) and the operating system. If communications or network software are used, the components of this should also be included or excluded if the network responsibility is another functional unit. The components comprising the hardware should be used such as disks, memory, type of central processing unit, add-in boards for the application or communications and any peripherals. Any documentation that is used with the CDS should be included in the configuration management system and listed in the log.
- The baseline configuration should be established at the installation of a new CDS, even if the system is used as a test environment before becoming operational. This has a number of advantages: first, all testing and training take place in a controlled environment and second, the procedures and principles of configuration management are known, understood and modified if necessary, before the system is rolled out for operational use. The information for the baseline configuration will come from the purchase order, and this will be checked off at the installation.
- Modifications to the system configuration can then be made and the information recorded in the configuration log or its equivalent. When new versions of the software are available and installed, master copies of the old version and the relevant documentation should be archived as they should be considered equivalent to raw data.

Documentation

Documentation can be divided into several categories, each will be discussed in this section. Excluded from discussion is the documentation produced during development and initial validation of the system that was discussed in detail in part III of this series (1).

System-specific documentation: The documentation supplied with the CDS application or system (both hardware and software), user notes and user standard operating procedures (SOPs) will not be discussed here as they are too specific and dependent upon the management approach in an individual laboratory. However, the importance of this system-specific documentation for validation should not be underestimated. Keeping this documentation current should be considered a vital part of ensuring the operational validation of any computerized system. The users should know where to find the current copies of documentation to enable them to do their job. The old versions of user SOPs, and system and user documentation should be archived.

Standard operating procedures: Standard operating procedures are required for the operation of both the CDS applications software and the system itself. As explained above, we have not considered user SOPs in detail. Standard operating procedures are the main media for formalizing procedures by describing the exact procedures to be followed to achieve a defined outcome. According to Hambloch (6), SOPs have the advantage that the same task is undertaken consistently, that it is done correctly and that nothing is omitted. In addition, a written procedure means that new employees are trained faster. The aim is to ensure a quality operation. Laboratory staff are used to working with SOPs; however, if a large system is supported by a central computer group they may not be used to working with SOPs and even less ready to document their work. To provide a service to a regulated laboratory, a computer department must provide a suitably documented procedure. Indeed this is a requirement under EU GMP Annex 11 (2), where a third-party supplier should have a documented operation.

According to Hambloch (6) there is a minimum list of 12 SOPs required for the operation of a computer system in a regulated or accredited laboratory. These are

- SOP on SOPs: this should describe the approach taken to the writing of SOPs within the functional group, the sections, who can authorize the procedure, description of the procedure and distribution list.
- Description of responsibilities: the roles and responsibilities of staff supporting the computer system are defined.
- System description of hardware and change-control procedures: describes how the hardware components will be maintained (equivalent to the hardware configuration log) with the procedure to be adopted when the system configuration is changed.
- Preventative maintenance: describes the procedures for preventative maintenance of the hardware components
- Prevention, detection and correction of errors: the measures and procedures for finding, recording and resolving errors in the system. This can be a complex SOP covering many different aspects of the system and may refer to sections of the technical manuals provided with the system. This SOP includes good housekeeping such as disk defragmentation or monitoring the space available on all disks.

- System boot and shutdown: this is a special SOP that should contain all the specific instructions for starting up and shutting down the system. This SOP may be required in an emergency and, therefore, should be well written and be easily available for use.
- Control of environmental conditions: For systems that require a controlled environment, an SOP should define the acceptable ranges of temperature, humidity and power supply. Other environmental considerations may be what to do in the situation of electrostatic discharges, power surges, fire, lightning strikes or the use and maintenance of an uninterruptable power supply (UPS).
- Contingency plans and emergency operation: this is a disaster-recovery plan and uses alternative plans until the computer system has been recovered. It is important that any disaster recovery plan is tested and verified before any disaster occurs.
- Back-up and restore of data: describes the procedures for back-up of data and software programs and how to restore data to disk.
- Security: the logical (software) and physical security of the system is covered with the procedures for setting up and maintaining security.
- Installation and update of software: procedures to be undertaken before, during and after installing software. This should start with the complete back-up of all disks and then installation of the software and any testing and validation that may be required.
- Development and update of system software procedures: software can be written to control the system or help execute functions. This SOP outlines the procedures for the creation, documentation and modification of these procedures.

The reader is referred to the article by Hambloch (6) for more details on these SOPs. However, it is important to realize that the list above refers to a relatively large computer system that is run by a centralized IT group. Therefore, for smaller items of laboratory computer equipment the list should be reviewed for applicability and suitability. Where a system does not have the facility to store raw data (e.g., a disk drive) then no SOP is required for back-up and restore. The same logic should be applied to the whole list. The converse is also true; this is a generalized list of SOPs and if there is a specialized application there may be the need for more SOPs than appear above.

Training records: All involved with the selection, installation, operation and use of the CDS should have and maintain training records to demonstrate that they are suitably qualified to perform their functions. It is especially important to have training records and curricula vitae of installers and operators of a system, as this is a particularly weak area and a system can generate an observation for non-compliance. Major suppliers of CDSs will usually provide certificates of training for installation of the system and software. However, a major weak spot with many CDSs that have the IT department running the system is that the personnel do not have the relevant training records or curricula vitae.

Training records for users are usually updated at the launch of a system but can lapse as a system becomes mature. To demonstrate operational control, training records need to be updated regularly and especially after software changes to the system. Error fixes do not usually require additional training. However, major enhancements or upgrades should trigger the consideration of additional training. The prudent laboratory would document the decision and the reasons not to offer additional training in this event.

To get the best out of the investment in a system, periodic retraining, refresher training or even advanced training courses could be very useful for large or complex ones. Again this additional training should be documented.

Operational Logbooks

To document the basic operations of the computer system a number of logbooks are required. The term logbook is used flexibly in this context; the actual physical form that the information takes is not the issue, rather the information that is required to demonstrate that the procedure actually occurred is. The physical form of the log can be a bound notebook, a pro-forma sheet, a database or anything else that records the information needed, as long as security and integrity of the records are maintained.

Typically, operations logs are required for back-up of data and program disks on a computer, recording errors of computer operation and their resolution and maintenance records for the system and its components. We will discuss each log in turn.

Back-up log: The aim of a back-up log is to provide a written record of data back-up and location of duplicate copies of the system (operating system and application software programs) and the data held on the computer. The back-up schedule for the disks can vary. In a larger system, the operating system and applications software will be separated from the data, which are stored on separate disks. The data change on a fast timescale reflects the progress of the samples through the laboratory and must be backed up more frequently. In contrast, the operating system and application programs change at a slower pace and are therefore more static; the back-up schedule can reflect this.

For smaller systems, such as personal computers, the data and programs may be located on the same disk and partitioned by the directory structure. If the back-up software is capable of performing selective back-ups then the comments in the paragraph above apply. However, if there is little sophistication the whole disk may have to be backed up routinely. Again, for PC systems this may be an area to evaluate closely before buying. An alternative is a PC network, where the programs and data are held on a central server and can be backed up more efficiently and effectively than stand-alone systems.

Some of the key questions to ask when determining the back-up of your CDS are

- How long should the time between back-ups be? This can be answered by considering how much data you can afford to lose. If it is up to a week, then the back-ups can be weekly. If you cannot afford to lose any data, shadowing or duplicate disks are the start of the answer that may lead you to consider RAID (Redundant Array of Inexpensive Disks) technology.
- Who is authorized to perform back-ups and who signs off the log? The laboratory manager in conjunction with the person responsible for the system should decide this. The authorization and any counter signature required should be defined in an SOP.
- When should duplicate copies be made for security of the data? This question is related to the security of your data and programs. Duplicate copies should be part of the back-up procedure at predetermined intervals. The duplicate copies should be stored in a separate location in case of a hazard to the computer, and the original back-ups should be located nearby. Duplicate back-ups are also necessary to overcome problems reading the primary back-up copies.

Problem recording and recovery: During the operation of a computer system, boot up, back-up or other system functions, it

will be inevitable that errors may occur. It is essential that these errors are recorded and the solutions to resolve them are also written down. Over time, this can provide a useful historical record to the operation of the computer system and the location of any problem areas in the basic operation.

Areas where this can occur may be in peripherals where a print queue has stalled. This is relatively minor; however, there may be situations where the application fails because of a previously undetected error. In the latter instance, there is a need to link the error resolution to the change-control system.

Software error logging and resolution: As mentioned previously (1), it is impossible to completely test all of the pathways through CDS software. It is inevitable that errors will occur during the operation of the system. These must be recorded and tracked until there is a resolution. Segalstad and Synnevag (7) have discussed errors and their resolution and there will be no detailed discussion here. The key elements of this process are to record the error, notify the support group (in-house or vendor), classify the problem and identify a way to resolve it.

Not all reported problems of a CDS will be resolved. They might be minor and have no fundamental effect on the operation of the system and may not even be fixed. Alternatively, a 'work around' may be required, which should be documented, even retraining may be necessary. Other errors may be fatal or major, which means the system cannot be used until fixed. In these situations, the revalidation policy will be triggered and the fix tested and validated before the CDS can be operational again.

Maintenance records: All quality systems need to demonstrate that the equipment used is properly maintained and in some instances calibrated. Computers are no exception to this. Therefore, records of the maintenance of the CDS need to be set up and updated in line with the work performed on it. The main emphasis of the maintenance records is towards the physical components of a system: hardware, networking and peripherals. The software maintenance is covered under the error logging system described above.

If the hardware has a preventative maintenance contract, the service records after each call should be placed in a file to create a historical record. Also, the occurrence of any additional problems that require maintenance will be recorded in the system log and there will need to be cross-references to the appropriate record there.

Many smaller computer systems have few if any preventative maintenance requirements, but this does not absolve the laboratory from keeping records of the maintenance of the system. If a fault occurs that requires a service engineer to visit, then this must be recorded as well.

On sites where maintenance of personal computers is performed centrally for reasons of cost or convenience, maintenance records may be held centrally. The remit of the central maintenance group may cover all areas of a site or organization including regulated or accredited as well as non-accredited groups. It is important for the central maintenance group to keep records that are sufficient to demonstrate to an inspector the work they undertake. As defined in EU GMP Annex 11 (2), the third-party undertaking this work should have a service agreement and also have the curriculum vitae of its service personnel available and up to date.

Audit trail: The integrity of data entered into a CDS must be maintained carefully as the electronic medium holding the data is less robust than the paper-based system it replaces. In practice this means that data, without proper controls and authorization, can be easily transformed or even lost by magnetic media and data security can be lower than with paper. It is incumbent upon the users of a system to ensure that data are not altered without proper authorization. In some systems, usually those built around databases such as LIMS and the newer CDSs, an audit trail is available.

In essence, an audit trail is a software utility that monitors changes to selected data sets within the main application. An audit trail is configurable, in that it can be decided which data sets are to be monitored. The reason for this configuration is that the use of an audit trail entails a processor overhead (i.e., you need more computing power to operate both the audit trail and the main application than the latter alone). Therefore, the data sets to be monitored should be those that have an impact on the integrity of the data, such as data acquired directly from instruments, results and supporting data, such as sample entry and release.

The audit trail must show who made the change, when they did it, what were the old and the new values (data are not erased) and why data were modified. This is required for paper systems. It is stated or indirectly implied for computer systems. Therefore, during the evaluation of a system it is important that the audit trail is evaluated for such features. When preparing the report or archiving the results for a sample, batch or study, the audit trail should be searched and the audit relating to the specific samples obtained and filed with the raw data and supporting information. This proactive approach can prevent many problems after a system is operational.

Revalidation Criteria

Any change to a CDS should trigger consideration of whether revalidation of the system is required. Note the use of the word "consider". There is usually a knee-jerk reaction that any change means that the whole system should be revalidated. One should take a more objective evaluation of the change and its impact before deciding whether full revalidation is necessary.

First, if revalidation is necessary, to what extent is it required to test: a software unit, module or the whole system? Thus, revalidation is defined by Chapman as "repetition of the validation process or a specific portion of it" (5). There may even be instances where no revalidation would be necessary after a change. However, the decision must be documented together with the rationale for it.

Therefore, a procedure is required to evaluate the impact of any change to a system and action taken accordingly. One way to evaluate a change is to review the impact that it would make to data accuracy, security and integrity, as outlined by Lepore (8). This will give an indication of the impact of the change on the system and the areas of the application affected. This allows you to target the revalidation effort that is appropriate to the change you are going to make.

Disaster recovery: Good computing practices require that a documented and tested disaster recovery plan must be available for all major computerized systems. It rarely is. Failure to have a disaster-recovery plan places the data and information stored by major systems at risk, with the ultimate losers being the workers in the laboratory and the organization itself.

Disaster recovery is usually forgotten, or not considered as "it will never happen to me". The recovery plan should have several shades of disaster documented. From the loss of a disk drive: how data will be restored from tape or back-up store and then updated with data not on back-up through to the complete loss of the computer room or building because of fire or natural disaster.

Once the plans have been formulated,

they should be tested and documented to see if they work. Failure to test the recovery plan will give a false sense of security and compound any disaster.

Conclusions

To maintain the validation status of a CDS system, both operational control on a day-to-day basis and effective change control must be established and effectively maintained. Both aspects are interrelated. Errors and features discovered during the operation or change requests initiated require change control. Once the change has been initiated, it can have an impact on the operational factors such as documentation or operational logs.

References

(1) R.D. McDowall, LC•GC Europe, 12(9), 568–576 (1999).
(2) Good Manufacturing Practice for Medicinal Products in the European Community, Annex 11, (Commission of the European Communities, Brussels, Belgium, 1997).
(3) Organization for Economic Co-operation and Development, (Consensus Document on Principles of Good Laboratory Practice to Computerized Systems, Paris, France, 1995).
(4) A.S. Nakagawa, LIMS: Implementation and Management, (Royal Society of Chemistry, Cambridge, UK, 1994).
(5) K.G. Chapman, Good Computer Validation Practices; Common Sense Implementation, T. Stokes et al., (Interpharm Press, Buffalo Grove, Illinois, USA, 1994) 47–74.
(6) H. Hambloch, Good Computer Validation Practices; Common Sense Implementation, T. Stokes et al., (Interpharm Press, Buffalo Grove, Illinois, USA, 1994) 113–140.
(7) S. Segalstad and M.J. Synnevad, Chemometrics and Intelligent Laboratory Systems: Laboratory Automation and Information Management, 26 (1994) 1–12.
(8) P.D. Lepore, Laboratory Information Management, 17, 283–286, (1992).

Bob McDowall is Visiting Research Fellow in the department of Chemistry at the University of Surrey, and Principal of McDowall Consulting, Bromley, Kent, UK. He is also a member of the Editorial Advisory Board of LC•GC Europe.