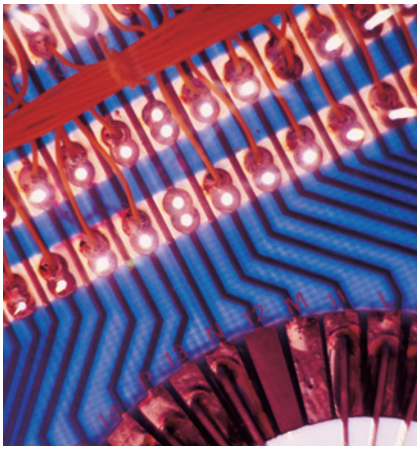
Technical Considerations for the

Validation of Electronic Spreadsheets for Complying with 21 *CFR* Part 11

Tuan T. Phan



DIGITAL VISION

Electronic spreadsheets are a valuable and pervasive industry tool but have yet to catch up with regulatory compliance demands without the implementation of an auxiliary program. From a historical perspective, the author discusses factors and strategies in this article for developing a spreadsheet validation plan.

Tuan T. Phan is president of Validation Associates, Inc., 9220 Sayornis Court, Raleigh, NC 27615, tel. 800.908.8230 or 703.855.3863, tphan@validationassociates.com, www.validationassociates.com.

This article presents a top–down approach to properly implementing a validation project for electronic spreadsheets. The article provides some interesting historical background about the evaluation of electronic spreadsheets and uses Microsoft Excel as the base spreadsheet application because it is widely used as such. The article discusses the shortcomings of electronic spreadsheets and highlights some unique solutions and techniques to help pharmaceutical companies comply with 21 *CFR* Part 11. The validation approach proposed herein is based on the good automated manufacturing practice (GAMP) methodology but also incorporates specific technical test cases and considerations that are unique to the qualification of spreadsheets to fully comply with the predicate regulations and 21 *CFR* Part 11.

Once upon a time in history

Electronic spreadsheets have been widely used in corporate America since their debut in 1978 with VisiCalc. The electronic spreadsheet evolution began with a Harvard Business School student, Daniel Bricklin, who needed an electronic spreadsheet for a case study report (1). With the help of an MIT acquaintance and later cocreator, Bob Frankston, the first electronic spreadsheet, VisiCalc, was born in the fall of 1978. Although VisiCalc's functionality was limited, more than one million copies have been sold to business users since 1979. In 1983, IBM's Lotus 1-2-3 software package emerged on the scene to assume the leadership role as the spreadsheet standard. Lotus 1-2-3 made using spreadsheets easier because it added integrated charting, plotting, and database capabilities. End users were also impressed with the program's complex calculation capabilities. Most important, Lotus 1-2-3 introduced naming cells, cell ranges, and spreadsheet macros, which not only remediated repetitive and periodic tasks, but also provided the tool to implement a new functionality to extend the underlying capabilities of a spreadsheet application.

Lotus 1-2-3 continued unabated until Microsoft Corporation emerged on the personal productivity market in 1984 with Excel, one of the first spreadsheets to use a graphical interface with pull-down menus and a point-and-click capability using a mouse pointing device. By late spring 1995, Excel had superseded Lotus 1-2-3 as the undisputed spreadsheet market leader. Microsoft has continued to maintain its dominance with enhancements to the capabilities of the Excel application, including the integration of a subset of Visual Basic programming language named Visual Basic for Applications (VBA) into the core Excel platform. This step allowed data sharing and manipulation within the application and within other Microsoft Office applications such as Access or Word.

Excerpts of warning letters

There are insufficient controls the integrity of calculated data generated by the [redacted] software in the quality controllaboratory, in that

- there is no audit trail to track the number of templates accessed to generate data calculations
- password protection can be bypassed in the system
- data files are automatically deleted after a hard copy is generated. There is no requirement to identify the analyst or time/date stamping of spreadsheet hard copies.

Failure to validate computer software used as part of the quality system for its intended use according to an established protocol as required by 21 *CFR* 820.70(i). For example, the data in the Excel spreadsheet identified as a "Hit List" of top non-conforming components contains 16 record counts for part number 8601618 DC converter failures compared to 18 record counts for part number 860168 DC converter failures in the DBASE database. The spreadsheet is used for management review of component suppliers for all components.

Your firm failed to validate several computer databases that are used for quality functions, including your Access database, your [redacted] software, and your MS Excel spreadsheet program as required by 21 *CFR* 820.70(i).

Failure to have an adequate validation procedure for computerized spreadsheets used for in-process and finished product analytical calculations: The current validation procedure uses only the values that result in within-specification findings, aberrant high findings, and aberrant low findings (21 *CFR* 211.165[e]). For example, SOP 644.00, QA/QC Spreadsheet Validation, is deficient in that only a small range of values are being used to challenge computerized spreadsheet mathematical calculations.

Constraints and remediations

From the beginning, validation of electronic spreadsheets has been a widely discussed topic among computer validation professionals as the programs found their way into laboratories, manufacturing plants, and regulatory applications. Today, possible applications of electronic spreadsheets include automating analytical data calculations from various laboratory analyses, tracking and summarizing product complaints, and gathering and summarizing clinical trial data collection and analyses. The validation effort, however, poses significant challenges primarily because of the capabilities of modern electronic spreadsheets. For example, Excel spreadsheet applications now can incorporate sophisticated user graphical interfaces with automated reporting and data manipulation and presentation through the use of forms, macros, and modules driven by high-level programming language such as VBA.

With the late-year 1997 introduction of 21 *CFR* Part 11–Electronic Records; Electronic Signatures, the complexity of validating electronic spreadsheets increased several folds (4). This regulation requires that electronic systems that manage regulated electronic records or use electronic signatures must be implemented with several key technical controls (2). Four key drivers against which systems must be evaluated are authenticity, data and system integrity, confidentiality, and nonrepudiation. Several warning letters have already been issued by FDA to highlight concerns based on these drivers (see sidebar, "Excerpts of warning letters") (7).

Unlike databases, electronic spreadsheets lack user-level security measures, which could render the application and the supporting data accessible to all users who gain access to the spreadsheet. This inherent weakness in authentication requires third-party solutions to close the security gap. One of the solutions, the use of a local area network and a network server, has already been deployed in many enterprises. By placing the electronic spreadsheet template onto the network (shared) directory, the organization can limit access to individuals by requiring the individual to log into the network and be assigned to an authorized user group. This approach also offers technical controls relating to user identification (ID) and password, including uniqueness of user ID, configuration of password, periodic expiration of user password, prevention of use of historical passwords, and so forth. A user-defined password also can limit access to protect the configuration of a spreadsheet, including cell contents, macros, and VBAs, from modification. The password must be maintained by a nonuser group within the organization, and password modification can be performed only under strict procedure controls.

Another critical but inherent limitation of electronic spreadsheet applications is the lack of an audit trail (5). This flaw allows data entries on the same spreadsheet to be altered by multiple users with no audit trail of what has transpired. Other weaknesses that have been cited by FDA in warning letters to companies include the inability to distinguish one saved version of an electronic file from another and the inability to authenticate hard-copy printouts against electronic spreadsheets from which the printouts purportedly have been generated (6). According to FDA, the lack of audit trails yields the greatest degree of compromise in data integrity.

Several off-the-shelf (OTS) solutions are available to minimize the degree of noncompliance with the requisite audit trails and data integrity. Some spreadsheet applications offer a change-tracking mechanism, but unfortunately these are not foolproof; they can be turned off (i.e., disabled) or falsified to suggest that changes are being implemented by another user. However, when the electronic spreadsheet is placed on a network server and is logged using the network auditing policy for accessing the file, a simple but effective audit trail against user access to the spreadsheet can be implemented. A data entry–level audit trail can also be implemented through the use of VBA or macros. However, this approach stipulates that users must access the application through graphical user interfaces, thereby allowing the entries or modifications to be logged against the default user name, which can be retrieved from the active network user profile. Care must be taken to ensure that the audit trail is captured and stored in such a way that prevents modification by users and that the execution of the programming codes is automated and cannot be bypassed. In addition, the programming codes must be safeguarded against modification by using a password protection option for modules and macros, and the availability of the password must belong to the developers and not the end users.

To authenticate a printout of an electronic spreadsheet, a hash generated from the data entered should be printed as part of either the header or footer of the printout. A National Institute of Standards and Technology (NIST)–compliant or approved hashing algorithm must be used such as the Secure Hash Algorithm SHA-1 from Federal Information Processing Standards FIPS (8). The generated hash must be based on the combination of the data entries stored and the date and timestamp of the saved electronic spreadsheet. The generated hash must be inserted into the printout using VBA codes only by executing the codes through the defined user interfaces. The codes should be structured so that bypassing the execution would render the printout unauthenticated (i.e. without the presence of the hash). A verification mechanism must also be provided within the application to allow the hash to be authenticated against the hash generated from the data stored within. In most cases, verification is simply implemented by requiring users to manually enter the hash obtained from the printout and compare the entered hash against the generated hash from the stored data entries of the electronic spreadsheet.

Other solutions that users can implement are available commercially. The electronic spreadsheet may be placed into an electronic document management system (EDMS) where the EDMS maintains the integrity of the spreadsheet and the data stored within. DaCS from Wimmer Systems is an example of an add-in for Microsoft Excel (10). According to the vendor, the DaCS add-in is designed to work with Excel to assure that 21 *CFR* Part 11 requirements are being met by automatically managing the audit trails, file security, and data integrity.

Electronic spreadsheets lack many of the technical controls that prevent them from becoming an easy target for data falsification. However, solutions are available that allow companies to comply with 21 *CFR* Part 11 requirements. When properly implemented with appropriate policies and procedures, these solutions, although not perfect, can demonstrate that the end users have taken reasonable steps to assure data authenticity, integrity, confidentiality, and nonrepudiation.

Preparing the user requirements specification

The user requirements specification, also known as the *functional requirements specification*, is a document that captures the functional requirements of the application. This document is typically prepared by the users to address the specific requirements they expect of the spreadsheet application and provide the basis for testing and qualification of the system. FDA states that the user requirements specification is fundamentally important to the validation of a computer system because it provides predetermined specifications from which a system should be operating (3). Unique considerations for spreadsheet applications that must be documented in the user requirements specification include

- types of data entries (text or numeric)
- range of data entries
- calculations
- functions
- reports
- graphs
- security, including spreadsheet, user, and data
- system performance, quality, error handling, start-up, shutdown, and so forth.

The types of data entry and the valid range of data must be clearly specified for several reasons, one of which is to satisfy the specific 21 *CFR* Part 11 controls such as sequence verification or device check. Examples of verification include

- that pH values entered are between 1 and 14
- that active ingredient concentration (purity) is between 0.0% and 100%
- the type of entry (text versus numeric)
- that numeric values are real or integer, positive only, nonzeros, and so forth
- that text values include date values that comply with specific configurations such as "mm/dd/yyyy" or are limited to a specific number of characters.

Calculations that are to be automatically performed by the spreadsheet must be documented as mathematic representation in a formula format along with the nomenclature and description of the input variables to be used. If boundaries are assigned to input variables, they must be explicitly specified for inclusion in the user requirements because the boundaries will help define the valid range of the data entries. An example of a mathematical representation of a calculation is shown in the following equation:

$$H = \frac{1}{\sqrt{2\pi\sigma^2}} \times \frac{e^{-(x-\mu)^2}}{2\sigma^2}$$

in which H is the height (ordinate) of a normal curve, μ is the mean, σ is the standard deviation, π is the constant 3.14159, e is the base of natural logarithms and is equal to 2.718282, and x can take on any value from $-\infty$ to $+\infty$.

Modern spreadsheet applications also can perform predefined calculations or operations using built-in functions. For Microsoft Excel, examples of some widely used functions include

- sum
- average
- stdev
- count
- if
- lookup
- round
- int.

The FDA guidance for the principles of software validation states that it is not appropriate to assume that the built-in functions of a spreadsheet application work as intended. The user requirements must specify what built-in functions will be used and in what context. If the vendor who supplied the spreadsheet application does not have adequate documentation to clearly define the mathematical expressions to be performed for the built-in functions, then the users must specify the mathematical representation for the built-in functions on the basis of the user's understanding of how the functions are to work. For complex functions that involve branching or the return of various results or calculations, the user requirements must document all possible outcomes of the function when it is evaluated (true and false outcomes). For functions that use lookup values, the lookup reference values, the lookup vector, and the resulting vector (altogether known as the *array values*), the outcomes must be documented in the user requirements specification. If extrapolation is to be used between values, the method must also be clearly specified.

The spreadsheet application may be required to generate specific reports and graphs using data collected and tallied from the users. The specific reports or graphs must be identified in the user requirements specification, including, at the minimum, the data to be summarized, any related data summary calculations, and the format in which the report or graph is to be presented (hard-copy printouts or electronic display).

Because spreadsheets inherently are not secure, the user requirements specification must address how the calculations, functions, reports, graphs, or programming codes (macros or VBA) are safeguarded against modification. Security measures will rely on a combination of physical and logical measures and are typically tiered into four levels to meet the requirements of 21 *CFR* Part 11 and predicated regulations. The four levels include facility, room, user, and function. At the facility level, access to the company site must be restricted to employees, and visitors must be escorted to or be limited to general-access areas. At the room level, access to the physical placement of the data servers where the master electronic spreadsheet resides is limited to specific employees and to a particular room or area. At both facility and room levels, security measures are typically implemented through combinations of lock and key, security guards and check-in desks, ID cards, and electronic-access cards. At the user level, security addresses the specific user group that may gain access to the electronic spreadsheet for use or modification. This security level involves the use of an access control list and the network security to be administered. The function level involves safeguards specific to functions within the application, including access to various levels of the application's menus based on the roles of the users. The function level may also be available through the custom implementation of forms, interfaces, and VBA.

The preparation of the user requirements specification is the responsibility of the end users. Care must be taken to ensure that the key characteristics of the user requirements element are

- unique
- concise
- unambiguous
- complete
- verifiable
- consistent
- understandable
- traceable
- design independent.

Utmost importance must be placed on the user requirements specification being as complete as possible before the design process is initiated. Because the document serves as the basis for qualifying the system, the document must be controlled and, therefore, it must be reviewed and approved by the appropriate personnel, who must include the end-user representative. Upon approval, the user requirements specification must be maintained under a change-control management process to prevent any unauthorized changes to the project and system development scope. This strategy will minimize disruptions to the system developers' efforts.

Documenting the design specification

With the requirements specification in hand, the users can next engage the developers to prototype the system. Once the final protocol type has been accepted by the users, the developers must submit a design specification outlining how the completed system will meet the requirements outlined in the user requirements specification. The design specification, at the minimum, must contain a description of the system, the design–development methodology used to create the system, key functions of the system, a requirements traceability matrix, and the specific technical infrastructure environment (for the client and server, the operating system, spreadsheet version, application version, etc.). A complete printout of source codes, including formulas within cells, custom codes such as macros and VBA, spreadsheet structure, and so forth, may also be included as part of the design specification along with the electronic data backup (or location where the backup is maintained).

As previously mentioned, the developers may rely on the integration of third-party solutions to meet some specific user requirements, particularly with 21 *CFR* Part 11 requirements. If used, the interfaces to such third-party packages must be clearly detailed and documented in the design specification. In addition, the individual third-party providers may be required to submit a design specification for the system, unless the solution is OTS, in which case the configuration of the user environment will be documented in the design specification. Vendor audits may also be required depending on how critical the application is to product and data quality.

The use of a requirements traceability matrix (see Table I) is helpful when determining if requirements have been properly addressed. Note that the same unique requirement can be addressed by one or more design elements in the design specification. However, it is less likely to have the same design element to address multiple requirements in a clearly written user requirements specification.

Similar to the user requirements specification, the design specification must also be controlled. The review and approval process of the document is typically determined by the vendor (developers); however, in most cases, that process will require the approval from the end user's technical representative.

Qualifying spreadsheet application

Following the GAMP methodology, computer systems are qualified according to their classification with the various software types. Table II presents a summary of the GAMP categories and the required qualification approach.

Table I: Requirements traceability matrix in design specification.

User Requirements Specification Element	Design Specification Element
REQ #1–3.1	13.2.1, 13.3, 15.1–15.3
REQ #2–3.2	12.1–12.7
REQ #3–3.3	13.2.1, 14.1–14.3

Accordingly, in most cases, electronic spreadsheets must be qualified within two GAMP categories. First, the application itself is considered to be a standard software package, or OTS, and thus is qualified as category 3. The operating system from which the application operates, such as UNIX or Windows 2000, must be qualified as category 1. However, if macros or VBA are used, the custom code portion of the application must be qualified as category 5. As previously stated, the vendor audit will depend on whether the company considers the application to be critical to its data and product quality as well as on the level of complexity of the custom codes.

The final guidance for the principles of software validation has reduced the requirements of installation, operational, and performance qualifications to simply testing to be demonstrated by applying predetermined testing procedures, or test cases, as defined by a preapproved test plan (a predefined written plan with a formal summary of testing and a record of formal acceptance). Documented evidence of all testing procedures, test input data, and test results must be retained. For every functional requirement element defined, at least one test case must be implemented. The test cases must include a confirmation of the system's hardware and software configuration. In addition, the test cases must also be developed so they will exercise the system to challenge its operations and performance against the predetermined criteria, especially for its most critical parameters. The test cases must be executed and the results must be recorded and evaluated to determine whether the results show that the software or application has been validated for its intended use. It may also be helpful to tie the test cases to the user requirements and to the design elements through a requirements traceability matrix as shown in Table III.

The qualification of a computerized system involves an examination of both the static and dynamic technical and nontechnical attributes and functions of the system. For nontech–nical verification, several verifications must be conducted, including procedure verification, safety verification, calibration, and preventative maintenance verification, to name a few. The details for these verifications are not discussed in this article because they are outside the article's scope. The technical portion of the qualification must begin with a static verification of the configuration of the environment. The hardware configuration of the server, where the spreadsheet application template resides, must be documented. The configuration of the client hardware from which the end users access the spreadsheet must also be recorded. Information that must be recorded includes computer model, type, manufacturer, serial number, and the appropriate device configuration such as memory, storage capacity, user interface devices (mouse, display monitor), peripherals (printer, modem, CD-ROM player and recorder), and so forth. For the software configuration, the version of the operating systems for the server and the clients, including any service pack upgrade, must also be documented. Specific to the clients, the version of the spreadsheet application from the vendor and the electronic spreadsheet template, which is assigned by the developers, must also be recorded. Other information that must be documented includes the name of the manufacturer of the software; the listing of the primary installation directory and the details of the files such as file name, file type, and timestamp; and the physical location of the installation media backup. If third-party solutions are used to ensure compliance with 21 *CFR* Part 11, the hardware and software configuration of the third-party solution must also be documented as part of the static verification.

The next phase of the static verification confirms that the formulas, macros, and VBA are not modifiable by the end users through ordinary means. For the qualification, a subset of the calculations and functions (formulas within cells) by column and by row is examined to confirm that, as implemented, they match the design specification. Care must be taken to select a verification sample that is large and representative of the spreadsheet's complexity. The sample size must also include the start, middle, and the end points because these cell locations are likely to have the greatest degree of mismatch against the remaining data set during a design copy session. The same analysis should also be repeated to other worksheets within the spreadsheet that are replicates of each other. At a minimum, at least three worksheets must be examined, with one at the first point, one at the middle point, and one at the last point of the set.

With the static verification completed, the dynamic verification can be executed to confirm the intended use of the spreadsheet. The basic operational verification of the spreadsheet application must be performed, including application start-up and shutdown to ensure that errors are not present. If the spreadsheet application is stored on the server, a data backup and restore operation must also be performed, unless the data server and the data backup–restore operation have already been validated. Other operational aspects such as report printing, template retrievals, and saving operations to the network server are also verified and documented. For networked systems in which the spreadsheet application template resides on a network server and is accessed from multiple network clients, the dynamic verification of the operational aspects must be confirmed for at least three different clients. The actual number of clients to qualify may be adjusted to represent the actual population. A good rule of thumb is the square root of the actual number of clients plus one.

The operational and performance challenges represent the essence of the qualification process and should be the last step to be taken. In this phase, predetermined data sets are applied against the spreadsheet to derive expected outcomes. The requirements of the spreadsheet are tested, as is the integration of the spreadsheet with any third-party solution. At least three different data sets are used for every testing scenario. The repeatability of the test scenarios with three data sets is intended to demonstrate that the spreadsheet application is capable of consistently meeting the intended behaviors.

The data sets selected for the qualification must test not only

Table II: Summary of software categories (9).

Category	Software Type	Qualification Approach
1	Operating system	● Record version, including any service pack upgrades.
2	Firmware	● For nonconfigurable firmware record version. Calibrate as necessary. Verify operation against user requirements. ● For configurable firmware, record version and configuration. Calibrate as necessary. Verify operation against user requirements. ● Manage custom firmware (Bespoke) as category 5.
3	Standard software packages	● Record version and configuration of the environment. Verify operation against user requirements. ● Depending on the critical nature and complexity of the application, consider auditing the supplier.
4	Configurable software packages	● Record version and configuration. Verify operation against user requirements. ● Normally audit the supplier of critical or complex application. ● Manage any custom (Bespoke) programming as category 5.
5	Custom (Bespoke software)	● Must audit supplier. Validate the complete system.

Table III: Requirements traceability matrix in qualification protocols.

User Requirements Specification Element	Design Specification Element	Test Case
REQ #1–3.1	13.2.1, 13.3, 15.1–15.3	TC #1, TC #2, TC #3
REQ #2–3.2	12.1–12.7	TC #4, TC #5
REQ #3–3.3	13.2.1, 14.1–14.3	TC #6, TC #7, TC #8, TC #9

values within the valid range of the functions and calculations but also at the valid range limits (high and low) and slightly beyond the range limits (above and below). For conditional branching functions, all branches and combinations thereof must be verified if complex branching is used by the function. For built-in functions, manually calculated and independently verified data sets should be used to confirm the accuracy of the built-in functions. If user interfaces (UIs) are made available through VBA, the elements of the UIs such as buttons, dropdown selection lists, option and text boxes, and so forth, are verified to ensure that the proper data interfaces have been implemented. Again, the same analysis should also be repeated to other worksheets within the spreadsheet that are replicates. At a minimum, at least three worksheets must be examined, with one at the first point, one at the middle point, and one at the last point of the set. However, unless the UIs use different code sets, the testing for UIs must be performed only once.

Documentation during the qualification process is an equally important phase. A combination of screenshots, reports, and printouts must be collected to serve as data evidence to support the qualification effort. A sufficient amount of data evidence must be collected to ensure that a third-party reviewer can derive the same conclusion as those who performed the actual qualification. The data evidence must be initialed and dated by the person who performed the qualification and be related to the specific step of the test case at which time the data evidence was collected.

The preparation of the final report is the final phase of the qualification effort. The final report summarizes the findings of the qualification. Deviations identified during the validation effort and their appropriate resolutions are discussed. A definitive statement must be made about whether the organization considers the validation of the spreadsheet application to be completed and whether the application meets the organization's policies and procedures for the intended use. The approval of the final report signifies system acceptance in that the application is ready for use and that change-control procedures will be used to control modification and requalification.

Conclusion

The use of spreadsheets pervades every functional area of an organization. Compliance with 21 *CFR* Part 11 was mandated to exercise regulatory control over computerized spreadsheet applications and the electronic records they produce. Unfortunately, modern spreadsheet design has yet to reach the necessary levels of sophistication for user control and data tracking as have other computerized applications such as databases. However, supplemental technological solutions are available to bring electronic spreadsheets toward compliance with 21 *CFR* Part 11.

To achieve compliance, end users must have a complete understanding of their spreadsheet application and how it is being used. End users must also understand the limitations of their electronic spreadsheet application and specifically what gaps exist. With a solid road map in place, the organization can then design a robust and compliant solution. The policies and procedures that users define for 21 *CFR* Part 11 compliance can also provide significant insights into what the organization must do and to what extent it must be done. In the end, when evaluating a system for compliance with 21 *CFR* Part 11 and the predicated regulations, the FDA inspector will only be concerned with the extent to which the key drivers such as authenticity, data and system integrity, data confidentiality, and nonrepudiation are addressed by the electronic spreadsheet as well as the underlying supporting infrastructure and that the proper qualification has been completed and documented.

References

1. D.J. Power, "A Brief History of Spreadsheets," http://www.DSSResources.com.
2. "21 *CFR* Part 11 Regulation—What Senior Executives Need to Know," Validation Associates, Inc., Raleigh, North Carolina, http://www.validationassociates.com.
3. FDA, Guidance for Industry, *General Principles of Software Validation,* 11 January 2002.
4. FDA, *21 CFR Part 11 Electronic Records; Electronic Signatures,* Final Rule, 20 March 1997.
5. FDA, Guidance for Industry, *21 CFR Part 11; Electronic Records; Electronic Signatures–Validation,* Rockville, Maryland, August 2001.
6. FDA, Guidance for Industry, *21 CFR Part 11; Electronic Records; Electronic Signatures–Timestamps,* Rockville, Maryland, February 2002.
7. FDA, *Enforcement Policy: Electronic Records; Electronic Signatures–Compliance Policy Guide: Guidance for FDA Personnel,* Rockville, Maryland, 1 July 1999.
8. US Department of Commerce, "Secure Hash Standard," *Federal Information Processing Standards,* Publication FIPS 180-1, 17 April 1995.
9. ISPE, "GAMP 4 Guide for Validation of Automated Systems," December 2001.
10. DaCS Product Information, Wimmer Systems, http://www.wimmersystems.com. **PT**

Circle/eINFO 43