SPC Explorer: Guidance for Regulated Industries

FDA Recommendations

The following excerpts are from *General Principles of Software Validation; Final Guidance for Industry and FDA Staff (http://www.fda.gov/cdrh/comp/guidance/938.html#_Toc517237968)*, issued January 11, 2002. The excerpts are not the complete text of the guidance, but were extracted to highlight relevant points.

> Software validation is a requirement of the Quality System regulation, which was published in the Federal Register on October 7, 1996 and took effect on June 1, 1997. (See Title 21 Code of Federal Regulations (CFR) Part 820, and 61 Federal Register (FR) 52602, respectively.) Validation requirements apply to software used as components in medical devices, to software that is itself a medical device, and to software used in production of the device or in implementation of the device manufacturer's quality system.
>
> In addition, computer systems used to create, modify, and maintain electronic records and to manage electronic signatures are also subject to the validation requirements. (See 21 CFR §11.10(a).) Such computer systems must be validated to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records.
>
> Off-the-shelf software may have many capabilities, only a few of which are needed by the device manufacturer. Device manufacturers are responsible for the adequacy of the software used in their devices, and used to produce devices. When device manufacturers purchase "off-the-shelf'' software, they must ensure that it will perform as intended in their chosen application. For off-the-shelf software used in manufacturing or in the quality system, additional guidance is included in Section 6.3.
>
> FDA considers software validation to be "**confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.**"
>
> Software verification and validation are difficult because a developer cannot test forever, and it is hard to know how much evidence is enough. In large measure, software validation is a matter of developing a "level of confidence" that the device meets all requirements and user expectations for the software automated functions and features of the device. Measures such as defects found in specifications documents, estimates of defects remaining, testing coverage, and other techniques are all used to develop an acceptable level of confidence before shipping the product. The level of confidence, and therefore the level of software validation, verification, and testing effort needed, will vary depending upon the safety risk (hazard) posed by the automated functions of the device.
>
> For many years, both FDA and regulated industry have attempted to understand and define software validation within the context of process validation terminology. For example, industry documents and other FDA validation guidance sometimes describe user site software validation in terms of installation qualification (IQ), operational qualification (OQ) and performance qualification (PQ). While IQ/OQ/PQ terminology has served its purpose well and is one of many legitimate ways to organize software validation tasks at the user site, this terminology may not be well understood among many software professionals, and it is not used elsewhere in this document.

Software developers should establish a software life cycle model that is appropriate for their product and organization. The software life cycle model that is selected should cover the software from its birth to its retirement. Activities in a typical software life cycle model include the following:

- Quality Planning
- System Requirements Definition
- Detailed Software Requirements Specification
- Software Design Specification
- Construction or Coding
- Testing
- Installation
- Operation and Support
- Maintenance
- Retirement

Verification, testing, and other tasks that support software validation occur during each of these activities.

As applied to software, the term maintenance does not mean the same as when applied to hardware. The operational maintenance of hardware and software are different because their failure/error mechanisms are different. Hardware maintenance typically includes preventive hardware maintenance actions, component replacement, and corrective changes. Software maintenance includes corrective, perfective, and adaptive maintenance but does not include preventive maintenance actions or software component replacement.

Changes made to correct errors and faults in the software are corrective maintenance. Changes made to the software to improve the performance, maintainability, or other attributes of the software system are perfective maintenance. Software changes to make the software system usable in a changed environment are adaptive maintenance.

When changes are made to a software system, either during initial development or during post release maintenance, sufficient regression analysis and testing should be conducted to demonstrate that portions of the software not involved in the change were not adversely impacted. This is in addition to testing that evaluates the correctness of the implemented change(s).

The specific validation effort necessary for each software change is determined by the type of change, the development products affected, and the impact of those products on the operation of the software. Careful and complete documentation of the design structure and interrelationships of various modules, interfaces, etc., can limit the validation effort needed when a change is made. The level of effort needed to fully validate a change is also dependent upon the degree to which validation of the original software was documented and archived. For example, test documentation, test cases, and results of previous verification and validation testing need to be archived if they are to be available for performing subsequent regression testing. Failure to archive this information for later use can significantly increase the level of effort and expense of revalidating the software after a change is made.

In addition to software verification and validation tasks that are part of the standard software development process, the following additional maintenance tasks should be addressed:

- **Software Validation Plan Revision -** For software that was previously validated, the existing software validation plan should be revised to support the validation of the revised software. If no previous software validation plan exists, such a plan should be established to support the validation of the revised software.

- **Anomaly Evaluation** – Software organizations frequently maintain documentation, such as software problem reports that describe software anomalies discovered and the specific corrective action taken to fix each anomaly. Too often, however, mistakes are repeated because software developers do not take the next step to determine the root causes of problems and make the process and procedural changes needed to avoid recurrence of the problem. Software anomalies should be evaluated in terms of their severity and their effects on system operation and safety, but they should also be treated as symptoms of process deficiencies in the quality system. A root cause analysis of anomalies can identify specific quality system deficiencies. Where trends are identified (e.g., recurrence of similar software anomalies), appropriate corrective and preventive actions must be implemented and documented to avoid further recurrence of similar quality problems. (See 21 CFR 820.100.)

- **Problem Identification and Resolution Tracking -** All problems discovered during maintenance of the software should be documented. The resolution of each problem should be tracked to ensure it is fixed, for historical reference, and for trending.

- **Proposed Change Assessment -** All proposed modifications, enhancements, or additions should be assessed to determine the effect each change would have on the system. This information should determine the extent to which verification and/or validation tasks need to be iterated.

- **Task Iteration -** For approved software changes, all necessary verification and validation tasks should be performed to ensure that planned changes are implemented correctly, all documentation is complete and up to date, and no unacceptable changes have occurred in software performance.

- **Documentation Updating** – Documentation should be carefully reviewed to determine which documents have been impacted by a change. All approved documents (e.g., specifications, test procedures, user manuals, etc.) that have been affected should be updated in accordance with configuration management procedures. Specifications should be updated before any maintenance and software changes are made.

SPC Explorer Documentation

With regard to the FDA recommendations, the SPC Explorer software is considered off-the-shelf software used in manufacturing or in the quality system. The software and its requirements are designed internally and not subject to the particular specifications of any customer, or use in a particular application. The user Help System provides the detailed descriptions of its intended functionality and installation requirements. Its primary purpose is the analysis of process data for statistical control and (in some cases) the storage of the data used in the analysis.

The SPC Explorer software was initially released in January, 1998, after undergoing thorough verification and validation processes. It uses Quality America's QA-ActiveSPC charting control for its statistical analysis and chart displays. The QA-ActiveSPC control was first released as a stand-alone product in September 1997, although the basis for its statistical analysis was developed for the initial release of our SPC-PC IV software in June 1991, which in turn was based on our DOS-based software dating back to 1993. As such, the volume of verification and validation exercises performed on its interface, and most importantly its charting analysis, is vast.

Given its years of use in various industries, SPC Explorer is clearly in the maintenance phase of the FDA's software life cycle model. The vast majority of its current development activities relate to perfective maintenance, a small portion to corrective maintenance, and an even smaller portion to adaptive maintenance.

The following steps outline the verification and validation activities for perfective, corrective and adaptive maintenance:

1. For each suggested change, a Functional Specification is constructed.
    a. The Functional Specification is documented within the development database and assigned to a specific release of the application.
    b. The Functional Specification outlines the general requirements for the change, and may also include how the change relates to any existing functionality of the software.
    c. The Functional Requirements for the software are provided in the user Help System. Functional Specifications for given changes to the software are aligned with the Functional Requirements.
2. The Functional Specification is reviewed by the Programmer assigned to the task.
    a. The Programmer documents a design strategy in the development database, and implements a Code Change that addresses the needs of the Functional Specification.
    b. When the change is necessitated by a program error (corrective maintenance), the source of the error is determined and documented as part of the Code Change record within the development database.
    c. The Functional Specification may be revised to meet general coding requirements, or for simplification of the application interface.
    d. Design and Code Changes initiated by junior programmers are reviewed by their supervision. If the Code Change does not meet the requirements, or if the implementation method is flawed, the Code Change is revised prior to Systems Testing.
    e. When applicable, recommendations for systems level testing of the software are documented by the Programmer and/or Reviewer.
3. Help System documentation is updated, where necessary, based on the software change.
4. System Testing is performed by qualified staff to ensure conformance to the Functional Specification.
    a. Results are documented in the development database and/or stored in the secure testing results directory on the network server and referenced in the development database.
    b. When Code Changes may influence integrity of data storage or output actions of the software, testing includes verification relative to previous results (regression testing).
    c. When Code Changes influence reported statistics, System Testing includes verification of relevant calculations using secondary methods, including manual or automated calculations, and/or comparison to other software applications.
    d. Results of System Tests are reviewed relative to the Functional Specification and knowledge of the software application by a senior member of Technical Services or Software Development to ensure that adequate testing has been performed.

SPC Explorer 21 Part 11 CFR Guidelines

CFR Item	Description	Software Feature	Notes to Administrator
Subpart B--Electronic Records Sec. 11.10 Controls for closed systems a.	Persons who use closed systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, when appropriate, the confidentiality of electronic records, and to ensure that the signer cannot readily repudiate the signed record as not genuine. Such procedures and controls shall include the following: Validation of systems to ensure accuracy, reliability, consistent intended performance, and the ability to discern invalid or altered records.	Each release is thoroughly tested to ensure accuracy of data analysis, and reliable, consistent performance of the user interface. Data may be marked as invalid, or altered, with appropriate reports provided to identify these records (see below).	Internal policies and procedures should be developed and documented to provide specific instruction on the use of the software, or selection of features within the software, to assure consistency of use.
b.	The ability to generate accurate and complete copies of records in both human readable and electronic form suitable for inspection, review, and copying by the agency. Persons should contact the agency if there are any questions regarding the ability of the agency to perform such review and copying of the electronic records.	Data and its analysis results may be viewed, exported, or printed based on a number of user-defined query criteria. Data may also be available via queries through third-party software, such as Crystal Reports.	
c.	Protection of records to enable their accurate and ready retrieval throughout the records retention period.	Data is stored in a secure relational database, using one of a number of database formats (MS Access, SQL Server, MYSQL).	Data may be archived following the record retention period established by the local policy and procedures.
d.	Limiting system access to authorized individuals.	Password-protected login is required for access.	
e.	Use of secure, computer-generated, time-stamped audit trails to independently record the date and time of operator entries and actions that create, modify, or delete electronic records. Record changes shall not obscure previously recorded information. Such audit trail documentation shall be retained for a period at least as long as that required for the subject electronic records and shall be available for agency review and copying.	Data is stored using the network date/time at the point of data entry. Subsequent deletion or editing of the data, including the date/time field, value or traceability item associated with the data record, is stored within the audit trail. The audit trail provides both the previous and the current value of the affected field.	
f.	Use of operational system checks to enforce permitted sequencing of steps and events, as appropriate.	Where appropriate, system checks verify that necessary fields are completed and relevant before data is committed to the database.	

g.	Use of authority checks to ensure that only authorized individuals can use the system, electronically sign a record, access the operation or computer system input or output device, alter a record, or perform the operation at hand.	Each relevant function within the database, including login, is access-controlled.	Access control privileges should be defined and applied for classes of users.
h.	Use of device (e.g., terminal) checks to determine, as appropriate, the validity of the source of data input or operational instruction.	Data may be entered via electronic instrumentation to ensure its validity. Data validity checks may also be defined for each characteristic.	
i.	Determination that persons who develop, maintain, or use electronic record/electronic signature systems have the education, training, and experience to perform their assigned tasks.	Software training is available.	Local policies and procedures should be defined.
j.	The establishment of, and adherence to, written policies that hold individuals accountable and responsible for actions initiated under their electronic signatures, in order to deter record and signature falsification.		Local policies and procedures should be defined.
k.	Use of appropriate controls over systems documentation including: (1) Adequate controls over the distribution of, access to, and use of documentation for system operation and maintenance. (2) Revision and change control procedures to maintain an audit trail that documents time-sequenced development and modification of systems documentation.	The system documentation is provided exclusively via the user Help System, which is not editable by the user community. Updates are provided, as necessary, with each release of the product, and are subject to change and revision control procedures. In addition, a Document change control interface is provided within the software to manage the revision and control of internal policies and procedures impacting system operation and maintenance.	Local policies and procedures may be revision controlled and available to specified users within the Document branch.
Sec. 11.30 Controls for open systems.	Persons who use open systems to create, modify, maintain, or transmit electronic records shall employ procedures and controls designed to ensure the authenticity, integrity, and, as appropriate, the confidentiality of electronic records from the point of their creation to the point of their receipt. Such procedures and controls shall include those identified in 11.10, as appropriate, and additional measures such as document encryption and use of appropriate digital signature standards to ensure, as necessary under the circumstances, record authenticity, integrity, and confidentiality.	Open systems are not supported or encouraged for industries subject to regulatory compliance.	Not recommended.

Sec. 11.50 Signature manifestations	a) Signed electronic records shall contain information associated with the signing that clearly indicates all of the following: (1) The printed name of the signer; (2) The date and time when the signature was executed; and (3) The meaning (such as review, approval, responsibility, or authorship) associated with the signature. (b) The items identified in paragraphs (a)(1), (a)(2), and (a)(3) of this section shall be subject to the same controls as for electronic records and shall be included as part of any human readable form of the electronic record (such as electronic display or printout).	All data entry and editing functions, when written to the database, include identification of the unique user login, the date and time of the entry or edit, as well as the specified value of any number of Traceability Fields, as well as Assignable Cause and Corrective Action fields, associated with the data. Electronic signatures, as the digitized form of a handwritten signature, are not stored as separate entities within the software.	
Sec. 11.70 Signature / record linking	Electronic signatures and handwritten signatures executed to electronic records shall be linked to their respective electronic records to ensure that the signatures cannot be excised, copied, or otherwise transferred to falsify an electronic record by ordinary means.	Handwritten signatures may be affixed to printed reports, and scanned as a digital image using third-party scanning software. These electronic documents may be stored within the database. As such, the software provides the facility to manage and control the access and updating of the documents. Editing of the electronic documents is not supported within the software.	
Subpart C--Electronic Signatures Sec. 11.100 General Requirements		As stated above, Electronic signatures, as the digitized form of a handwritten signature, are not stored as a separate field within the software, so this section may be inapplicable. However, there has been some interpretation of signatures as unique user logins, meant to differentiate and identify users.	
a	Each electronic signature shall be unique to one individual and shall not be reused by, or reassigned to, anyone else.	User logins are unique. New users may not use the same login as existing users.	Policies and procedures should prevent Administrators from re-assigning a login to a different individual,

b	Before an organization establishes, assigns, certifies, or otherwise sanctions an individual's electronic signature, or any element of such electronic signature, the organization shall verify the identity of the individual.		Internal policies and procedures required.
c	Persons using electronic signatures shall, prior to or at the time of such use, certify to the agency that the electronic signatures in their system, used on or after August 20, 1997, are intended to be the legally binding equivalent of traditional handwritten signatures. (1) The certification shall be submitted in paper form and signed with a traditional handwritten signature, to the Office of Regional Operations (HFC-100), 5600 Fishers Lane, Rockville, MD 20857. (2) Persons using electronic signatures shall, upon agency request, provide additional certification or testimony that a specific electronic signature is the legally binding equivalent of the signer's handwritten signature.	Does not apply to software that does not include provisions for electronic signatures as digitized form of handwritten signature.	
Sec. 11.200 Electronic signature components and controls a	Electronic signatures that are not based upon biometrics shall: (1) Employ at least two distinct identification components such as an identification code and password. (i) When an individual executes a series of signings during a single, continuous period of controlled system access, the first signing shall be executed using all electronic signature components; subsequent signings shall be executed using at least one electronic signature component that is only executable by, and designed to be used only by, the individual. (ii) When an individual executes one or more signings not performed during a single, continuous period of controlled system access, each signing shall be executed using all of the electronic signature components. (2) Be used only by their genuine owners; and	Access to the software may be restricted based on two unique identification codes (login ID and password). Passwords are confidential, encrypted within dialog boxes, and known only to the user. Electronic signings are not permitted within the software. Users must re-enter login information after a specified period (in minutes) of inactivity.	Internal policies and procedures should require users to maintain and secure passwords. Use the *Require User to Logon After Timeout* feature in the Preferences dialog to specify the allowed period of inactivity.

	(3) Be administered and executed to ensure that attempted use of an individual's electronic signature by anyone other than its genuine owner requires collaboration of two or more individuals.		
b	Electronic signatures based upon biometrics shall be designed to ensure that they cannot be used by anyone other than their genuine owners.	Does not apply to software that does not include provisions for biometric identification of users.	
Sec. 11.300 Controls for identification codes/passwords a	Persons who use electronic signatures based upon use of identification codes in combination with passwords shall employ controls to ensure their security and integrity. Such controls shall include: (a) Maintaining the uniqueness of each combined identification code and password, such that no two individuals have the same combination of identification code and password.	Access to the software may be restricted based on two unique identification codes (login ID and password).	
b	Ensuring that identification code and password issuances are periodically checked, recalled, or revised (e.g., to cover such events as password aging).	Users may be forced to verify/change passwords on a specified interval.	Use the *Password Expiration* feature in the Preferences dialog box to specify the expiration period.
c	Following loss management procedures to electronically deauthorize lost, stolen, missing, or otherwise potentially compromised tokens, cards, and other devices that bear or generate identification code or password information, and to issue temporary or permanent replacements using suitable, rigorous controls.	Administrators may make any individual's login access rights inactive for any period of time, as needed.	
d	Use of transaction safeguards to prevent unauthorized use of passwords and/or identification codes, and to detect and report in an immediate and urgent manner any attempts at their unauthorized use to the system security unit, and, as appropriate, to organizational management.	Email may be directed to a specified Administrator whenever a specified number of unsuccessful login attempts has occurred.	Use the *Restrict Access after Failed Attempts Expiration* feature in the Preferences dialog.
e	Initial and periodic testing of devices, such as tokens or cards, that bear or generate identification code or password information to ensure that they function properly and have not been altered in an unauthorized manner.		Internal policies and procedures required.