

E-Validation

A Method for Electronic Validation Protocol Generation, Approval, and Execution

Paul N. Schank* and Craig M. Torres



Paul N. Schank is the senior computer validation engineer at Lockwood Greene Engineers, Somerset, NJ, pschank@lg.com. **Craig M. Torres** is a validation engineer also at Lockwood Greene Engineers.

*To whom all correspondence should be addressed.

A new method for applying a familiar technology to validation procedures may help change what typically has been a paper-based process. This article outlines a method for **executing an all-electronic validation**, which, in the future, may cause validation to be considered a paperless function.

Imagine that you have just finished authoring a validation protocol. You electronically sign the document and sequentially email it to the various approvers who also electronically sign it and return it to you. After opening your last e-mail with the completely approved file, you send a copy of the file to the document-control server for archival, download another copy onto your tablet portable computer (using a wireless Ethernet connection), and proceed to the field where you begin to execute the protocol by typing right into the file.

To most validation-savvy people, this situation may sound like a preview of the not-too-distant future; however, this is a real-world scenario that is already occurring. This article presents a method for electronic protocol generation and execution called *e-validation* and describes the positive and negative factors associated with this method.

The ultimate benefits of e-validation are increasing the legibility of completed protocols and saving physical space in document-control centers that house binders full of validation papers. Conspicuously missing from these benefits are project time and cost savings because this type of e-validation has higher associated time and monetary costs than a typical validation.

The e-validation example described in this article requires that a traditional validation program is already in place and that a method for the generation, approval, and execution of standard paper-based protocols (i.e., installation, operational, and performance qualifications) is understood. The presented methods also require a basic knowledge of the Adobe Acrobat software. This article intends to describe a method for implementing e-validation and does not discuss the nuts and bolts of using the software.

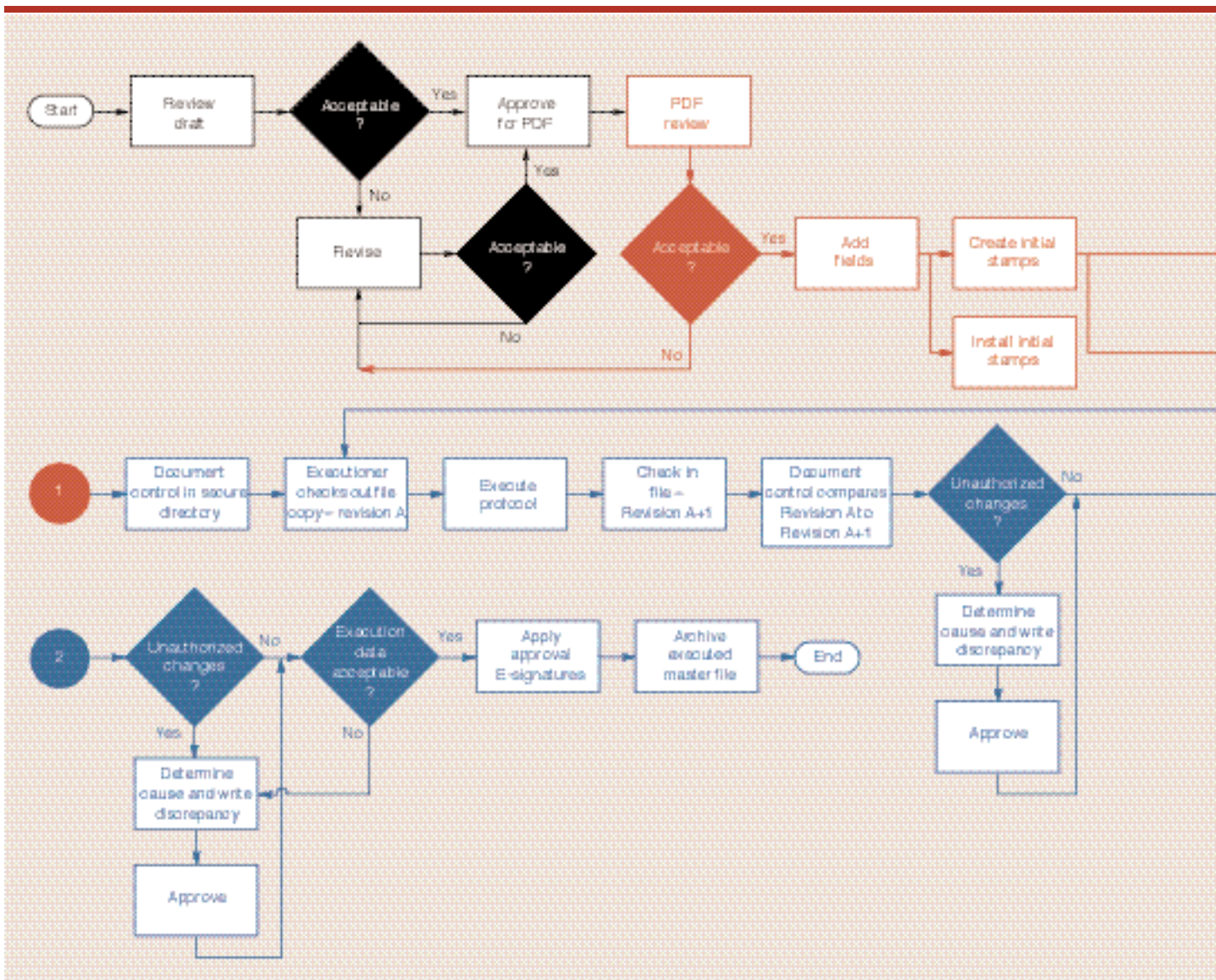


Figure 1: E-validation electronic protocol generation and execution flow chart.

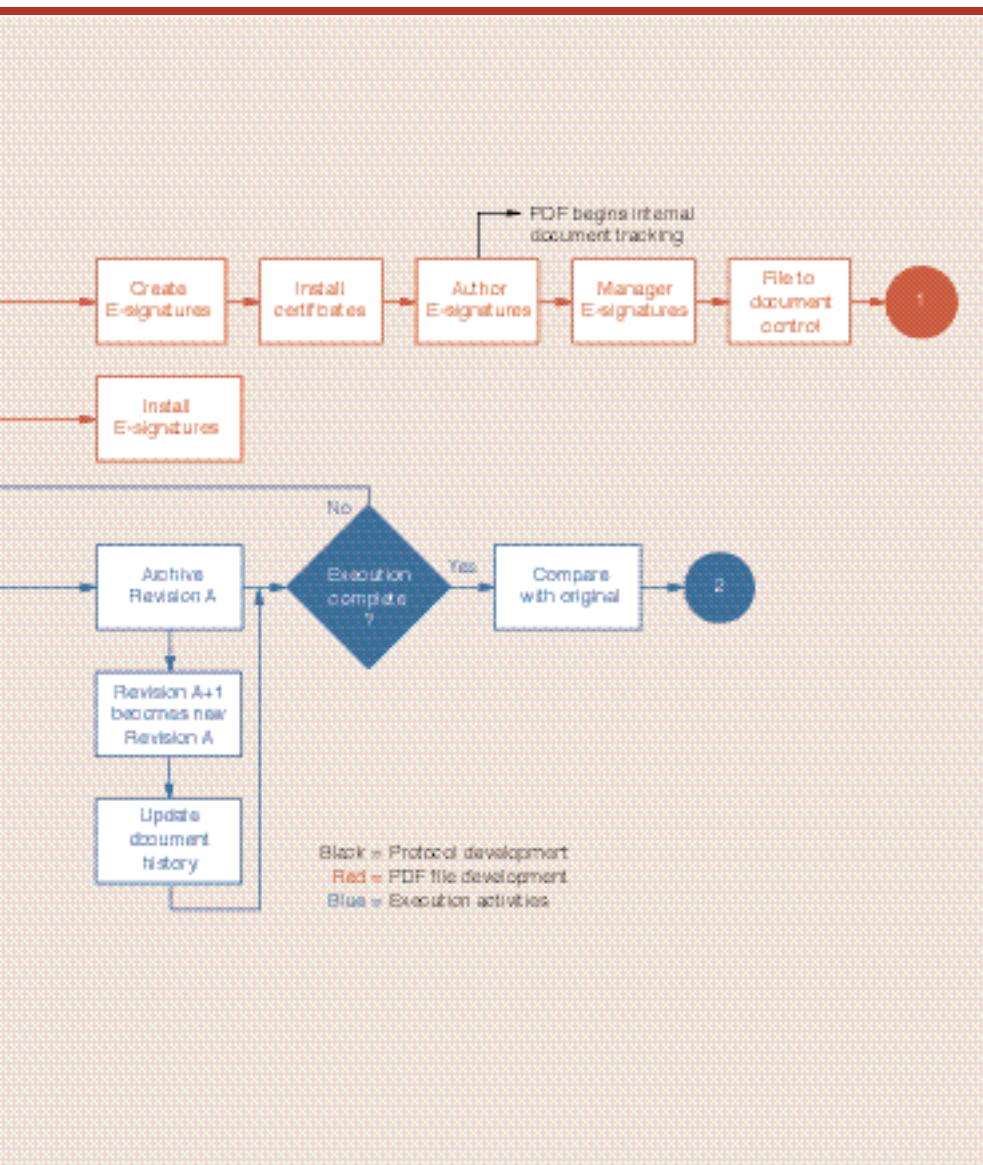
For this example, the authors chose the software program Adobe Acrobat to provide electronic signatures and audit trails in compliance with 21 *CFR* Part 11. Acrobat was chosen for its rich set of functions that complements federal regulations concerning electronic signatures, its ease of implementation, and, perhaps most important, its universal file format, the portable document format file (referred to as a *PDF*). This file format is widely used throughout industry and the world and can be viewed on any computer system running the Adobe Acrobat Reader software, which is available at no cost, although the Acrobat software used to create the PDFs must be purchased separately.

In addition, PDFs can be viewed in supported Web browsers in which Acrobat Reader functions as a plug-in to the browser. In the pharmaceutical industry, using PDFs for electronic documents has many advantages such as

- PDFs created on a personal computer or Macintosh platform are identical and require no conversion to be read cross-platform.
- Hundreds of government agencies (including FDA) have adopted the PDF format.
- Millions of PDF documents are posted on government Web sites.
- The federal government's E-Government Strategic Plan is published as a PDF.

Creating an executable PDF

The process of creating an executable PDF begins where the process of preparing traditional validation protocols ends (see Figure 1). Traditional protocols are created and undergo a typical review and editing process using standard word processing software (Microsoft Word was used for this example). Once the protocol has been completed electronically and is at the point where approval signatures would usually be applied to a hard copy, the document can be converted into a PDF file. This procedure is accomplished by a single mouse click on the conver-



task that is fairly labor intensive and time consuming.

After the unique smart fields have been placed in the document, the PDF can be submitted for a final execution review. The reviewer then applies his or her electronic signature, which begins the automatic tracking of any and all changes made to the PDF from this point forward.

Security: controlling access to PDF files

Many similarities exist between the procedures used to control paper documents and those used to control electronic documents. When the document, whether paper or electronic, is at the document-approval point in the process, the document-control unit becomes responsible for all access to the PDF.

The PDF should be located in a secure server location, and only the document-control unit should have access to this location, which is similar to a master paper copy being locked in a vault that is controlled by the document-control unit. Secure PDFs should be checked out through the e-document control (EDC) unit according to preapproved procedures, with the EDC unit keeping a log of all check-out and check-in activities. During check-out, the PDF can be loaded onto a portable computer and taken to the plant or laboratory for execution. PDFs should be checked out by only one individual at a time. Once execution activities are complete, the PDF can be checked in by the EDC unit according to approved procedures.

As part of the check-in procedures, EDC personnel should compare the checked-out PDF with the updated PDF that is being checked in. This comparison can be executed by Acrobat's "compare" feature and allows the EDC unit to readily determine if any unauthorized changes were made to the document.

As part of the check-in procedures, EDC personnel should compare the checked-out PDF with the updated PDF that is being checked in. This comparison can be executed by Acrobat's "compare" feature and allows the EDC unit to readily determine if any unauthorized changes were made to the document.

Executing a PDF

The document executioner must obtain a portable computer for execution purposes and check out the required PDF through the EDC unit in accordance with approved procedures. Several manufacturers now produce tablet portable computers, which are much easier to handle in the field than a typical laptop computer. Of course, when deciding which hardware to use in the field, one must weigh the pros and cons of carrying a laptop, which has a typewriter-style keyboard but is large and heavy, versus a tablet portable computer, which is easier to handle but uses a stylus and touch screen (similar to the Palm Pilot) for data entry.

sion button that Acrobat installs in all Microsoft Office applications. The document should be in a final-draft state before conversion because any significant changes to the document after it has been converted must be made using the original application (e.g., Microsoft Word), in which case the conversion must be repeated. Acrobat is essentially a post-processor that only allows simple document modifications to be made once the document has been converted to PDF.

For a paper document, the protocol would be created with empty boxes and blank lines that would be filled in by the documents executioner. After the file has been converted to a PDF, it should be thought of as an electronic picture of the original document in which the empty boxes and blank lines cannot be filled in. The next step is to insert or overlay text and signature fields in the proper locations to transform the PDF into a file that can be executed. Adding these "smart" fields provides the necessary interactivity to allow the executioner to fill in the blanks electronically and is one of the key functions of performing compliant e-validation. It is important to note that each field placed in the document must be assigned a unique field identifier—a

When executioners are in the field, they must log in to Acrobat using their unique self-sign security user identification and password. Once the PDF is opened, clicking on and typing in the form fields will record the execution activities in the PDF. All entries can be initialed and dated by applying an electronic initials stamp (another function of Acrobat) and typing the date in the appropriate fields.

For validation protocols, the executioner must apply his or her electronic signature in the locations provided after completing a protocol page or series of pages. Typically, once a page has been signed, it is then reviewed by a designated reviewer. If the page is deemed satisfactory, the reviewer applies his or her electronic signature in the locations provided.

After the executioner's activities have been completed (the document does not require full execution), the document should be checked in by the EDC unit in accordance with approved procedures. This system of execution and approval continues until protocol execution is complete and all reviewers have electronically signed and approved the protocol, which ultimately is archived by the EDC unit.

Using electronic signatures

Using electronic signatures comprises two functions: applying a signature and verifying a signature.

Applying an electronic signature. First, EDC personnel and the user must create an electronic signature file. The signature file may contain information about the user such as name, company, position, and contact information as well as the user's password and a unique key string that is used to authenticate any signatures applied by the user.

The electronic signature file must be installed in a password-protected folder on each computer that an executioner may potentially use for execution. Only EDC personnel (or the person they have designated) should have access to this folder. To apply an electronic signature, the user must click on a signature field, enter the reason for applying the signature, and enter his or her password.

Verifying an electronic signature. Electronic signatures in PDFs are verified by certificates. Although certificates can be extracted from electronic signatures, the most secure method of signature verification is to access certificates from a secure server location. As a result, no doubt should exist about an electronic signature's authenticity.

Clicking on an electronic signature (or choosing "verify signature" from the menu) will prompt the person verifying a signature to log in to the Acrobat self-sign security function to access the verifier's list of trusted certificates. After the user has logged in, the electronic signature embedded in the PDF is compared with the certificate found in the list of trusted certificates. If the electronic signature is found to be authentic, it is marked as valid.

Controlling electronic signatures

The same procedural rules governing computer-system passwords also apply to electronic-signature passwords. Users must select passwords that are not easily guessed (preferably a combination of numbers and upper- and lower-case letters) and

must never give them out. Although the EDC unit controls access to the signature files and verification certificates, it does not have access to user passwords. This procedure is similar to the way that Microsoft Windows NT passwords are administered.

After a user creates a signature, it is associated with a profile. Each profile contains two keys: a public key and a private key. The private key is used to sign a document. When a user attaches a signature, the public key is embedded in the signature and can be used later to verify the signature. For other users to verify the signature, they must have access to the public key, which is contained in a certificate. Adobe Acrobat's password protection function is based on a private- and public-key system using an RSA 1024-bit key algorithm, which supplies an adequate level of security for even the most conservative organizations.

Preventing and tracking document changes

Similar to paper documents, a copy of the original approved document is kept in a secure location for comparison with the executed version. After the original protocol document is converted to a PDF and the first electronic signature (usually the author's) is applied, Acrobat begins an internal-revisioning process. Each time the document is saved (or a signature is applied), a new internal revision is initiated, and all internal revisions are kept as part of the living document. As a result, any and all changes between internal revisions are tracked and easily identified.

Internal revisions may be compared with each other (by using Acrobat's "compare" function) to identify the nature of each revision's changes. Acceptable changes among internal revisions include merely filling out blank fields or adding additional deviation pages; however, unacceptable changes include any number of unauthorized changes to the document. Basically, all changes to a document are tracked and readily available for viewing. All documents should be reviewed and compared before any final electronic approval signatures are applied.

Data entry errors. In the case of a minor change (e.g., a typing mistake or data entry error in a validation protocol), an explanation in the "comments" section is a sufficient solution. In the case of a major change (e.g., a piece of equipment does not match the specification listed in the protocol), a deviation report is required to accompany the document change.

Required standard operating procedures (SOPs)

Although PDFs provide the necessary tools to enable electronic validation protocol executions, PDFs by themselves are insufficient to enable a QA-sanctioned electronic execution that is consistent with 21 *CFR* Part 11. The following procedures (which likely already exist for paper-based validations) must be developed and put into practice before most QA units will approve an e-validation:

Examples of document-control procedures are

- electronic-signature file maintenance
- electronic-signature certificate maintenance
- PDF check-in and check-out procedures
- PDF revision comparisons
- initials-stamp creation and maintenance.

Examples of electronic execution procedures are

- file check-in and check-out procedures
- entering data
- adding form fields
- adding deviation pages
- using and verifying electronic signatures
- using initials stamps.

Conclusion

The United States is in the midst of a great migration toward a paperless society, and our industry is not immune. Make no mistake, real costs are associated with regulation-compliant e-validation that are higher than the current cost of paper-based validations. More work and scheduled time are required to prepare a protocol for electronic execution. Current technology does not offer a way around the extra effort, yet. In addition, new procedures must be developed and implemented in an organization before e-validation can be adopted in a regulation-compliant manner. Apart from these negatives, certain problems exist that only e-validation can address.

The amount of storage space needed for paper-based validations is a very real and costly concern. Huge volumes of paper that are generated during a typical validation campaign must be securely stored somewhere once the project is completed. An electronic validation greatly reduces required physical storage space and its associated costs. As one can imagine, a single

shelf of CD ROMs easily replaces rooms that are full of binders. E-validation can reduce storage costs and free up much-needed space in an organization for other endeavors such as new R&D laboratories or more office space.

The legibility of executed protocols is an equally important consideration. The old validation adage, "If it wasn't written, it wasn't done" might as well be reworded, "If it wasn't written legibly, it wasn't done." After a project's execution activities are completed, the only record of them ever having occurred is the executed protocol. A company's validation-testing methods may be top-notch, but if an executed protocol cannot be deciphered, the document is useless, and the time, effort, and money spent on the validation is wasted. When viewed in this light, the cost of e-validation compares favorably with the costs associated with reexecuting illegible validation protocols (and a possible failed finished product).

E-validation is still in its infancy, but we as an industry are heading in this direction. Although the costs associated with developing electronically executable protocols are significant, they are essentially offset by savings in storage costs and assurance of legibility. As the industry increasingly begins to incorporate e-validation, software developers will no doubt begin to make it easier and less expensive. **PT**