Validation of Commercial-Off-the Shelf (COTS) Software

By George N. Brower
Analex Corporation

In the last few years, especially since the FDA's new Quality System Regulation was finalized in 1997, and with the phase-in of the design control regulations June 1, 1998,[1] many individuals in the medical device and pharmaceutical industries have been confused about the requirements for validation of Commercial-Off-the-Shelf (COTS) software. The questions frequently asked range from, "Is validation required?" to "How does a company even begin to do COTS software validation?" This paper discusses COTS software in general and which COTS software must be validated specifically. The FDA's requirements for validation are itemized, followed by a description of an approach to the task of software validation for the various types of COTS software. Suggestions as to how the costs may be mitigated are presented and, finally, this paper discusses vendor auditing and what to do if a vendor does not allow an audit when one is called for.

"This paper discusses COTS software in general and which COTS software must be validated specifically."

What is Commercial Off-The-Shelf Software?

COTS software is developed and sold for general use. That is, it is neither developed nor tailored for a particular user. The user does not have complete life-cycle control over the software. For example, if the developer decides to issue a new release (or, conversely, to not issue a new release) without regard to the user, the user does not have life-cycle control over the software. It should be recognized that COTS software whose source code is uniquely altered for a user is no longer strictly considered COTS software. The reason for this is any benefits that may be claimed for COTS software through wide-spread use, benefits ranging from known behavior to known problems (i.e., bugs), are lost as soon as the code is tailored for a single user.

Once and for All, Does COTS Software Require Validation, and Why?

The FDA's view on software validation is: Manufacturers have the ultimate responsibility for the software they use, whether the software is developed in-house, by a contractor, or purchased from a vendor; Manufacturers cannot assume current Good Manufacturing Practices (cGMP) were followed for the development or validation of the software – documented evidence of cGMP is required; and finally, the Quality System Regulation, especially 21 CFR Part 820.30, Design Control (fully in effect June 1, 1998),[1] must be followed. The FDA's Statement of Principles includes: quality, safety, and effectiveness must be designed and built into the product; quality cannot be inspected or tested into the finished product; and each step of the manufacturing process must be controlled to maximize the probability that the finished product meets all quality and design specifications. None of the foregoing states "except for COTS software." So, yes, COTS software must be validated.

But the FDA aside, following cGMP for software validation is the best approach for a user of COTS software in any case, because COTS software validation is a plus to the user in and of itself. This is so because validation supports the successful use and maintenance of the software by stabilizing the use through the identification of the setup, identifying individual problems and solutions quickly due to the small scope of each validation step, enhancing updates to the software and to any configuration files by verifying the initial completeness and uniformity of the setup, and supporting the cost and schedule estimation for management. If validation planning and execution start early, COTS validation can have a positive impact on the use of the product. However, oftentimes companies may need a reminder of all of the above. Whereas it is legitimate to ask, "What will be the cost of the COTS software validation?" a company may need to be reminded of what the possible cost will be if no validation is done.

Which COTS Software Must be Validated?

Per the FDA, cGMP requirements apply to software used in medical devices, manufacturing processes, and in a manufacturer's quality system. Furthermore, there are no fundamental differences in software validation between the above systems; all validation is primarily contingent upon the possible risk to a patient or healthcare provider should the software be deficient as a result of failure, malfunction, or misuse. However, there are COTS software validation challenges. For example, COTS software source code and development documentation may be unavailable to the user's validation team, because the data may be considered proprietary. Furthermore, COTS software is often so extensive, a complete software validation effort is not practical due to both cost and schedule prohibitions. But there are solutions to the COTS software validation problems. The FDA requires only the appropriate level of validation, with the results of the risk analysis being the major input. One caution, however, is if a major hazard cannot be mitigated due to the inability to audit the vendor's development, the COTS software may not be appropriate for use. Otherwise, the software, at a minimum, should be validated specifically for its intended use. An explanation of "validate the COTS software for its intended use" may be illustrated with spreadsheet software, where the user's Standard Operating Procedure (SOP) may state, e.g., "If the values in cells G32 to G35 are between 5.1 and 10.0 the process is under control and may continue." In this instance, only the spreadsheet form need be validated. The bar chart or pie chart capabilities need not be validated if they are not specified for use in any SOP.

As background information, *Software Validation*, as described by the FDA in the General Principles of Software Validation,[2] covers the whole life cycle of the software development process. However, software validation may be thought of as an umbrella encompassing both *software verification* and *software validation* as defined in the FDA's Quality System Regulation[1] and as used in the FDA's Guidance for Off-the-Shelf-Software Use in Medical Devices[3] and Guidance for the Content of Premarket Submissions.[4] As defined and used in those guidance documents, software verification confirms that the output of each software development phase is consistent with the inputs to that phase. Software validation confirms that the final software program, running in its intended hardware and environment, is consistent with the intended use of the product as defined primarily in the product specifications and supplemented, as needed, by the software requirements. This concept of software verification and validation is used in the succeeding sections. To differentiate between the larger meaning of software validation (i.e., one that encompasses both verification and validation), that term is capitalized hereafter, whereas in the narrower use of software validation (i.e., as in verification and validation), as defined in the FDA's Quality System Regulation,[1] validation is not capitalized.

What is the Difference Between Configurable and Nonconfigurable COTS Software?

COTS software can be categorized (i.e., typed) as configurable and nonconfigurable. Configurable software is a product that gives the user control over its internal functioning without the need for altering the source code. Rather, there may be a text configuration file or other keyboard text input, a "select and click" configuration capability, or a combination thereof.

Two examples of configurable COTS software products are a spreadsheet such as Excel running on a personal computer and an analog-to-digital (A/D) input software driver running in an embedded microprocessor of a medical device. Both of these must be validated for their intended use and, being configurable, they would both normally require verification in the area of user configuration, that is, verification that the configuration is consistent with the requirements for the configuration. For the spreadsheet, the configuration may be in the form of equations in cells to get a summation or a data average, and for the A/D driver, the configuration may be in the form of a configuration file dictating the data collection rate and the input device number assigned to each channel.

Examples of nonconfigurable COTS software may be an operating system such as Windows 95 and a word processor such as Microsoft Word. While these two examples both have a flavor of configuration (e.g., the background screen can be altered in Windows 95, and the font style can be selected in Microsoft Word), the results of these types of configurations are readily seen by the user and, thus, do not have hidden results (whereas defining a cell in Excel as containing a "mean" rather than an "average" may remain unnoticed by the user). Perhaps more importantly, these minor "configurations" do not fundamentally change the outcome of the use of the software. That is, if a report is written using a Courier font rather than a Times New Roman font, the meaning of the printed word is not changed.

Nonconfigurable software must also be validated for its intended use. However, especially in the case of nonconfigurable software, the concept of "verification by output" may be sufficient. Verification by output for a COTS software automated process or quality system function is acceptable to the FDA if the output of the process or function can be fully verified in every case against the specifications, in which case the process or function is not considered dependent upon proper operation of the software.

How Should the Validation of Each Type of COTS Software be Approached?

Most COTS software verification and validation guidelines are related to medical devices. More guidelines are expected to be forthcoming from the FDA. However, the present guidelines can be applied to all COTS software to the extent they are applicable. As with all software that is intended to meet the FDA's Quality System Regulation, the first question to ask for COTS software is "What is the possible risk to the patient or healthcare provider if the software has a deficiency?" An approach to answering this question is by performing a hazard (or risk) analysis.

Hazard Analysis, Mitigation and Conclusion

The medical device manufacturer is expected to perform a COTS hazard analysis[3] (a.k.a. risk analysis) as a part of the medical device hazard analysis, which should extend throughout the life cycle of product. However, as stated above, this should be a significant consideration for all COTS software and systems. There are many approaches to performing a hazard analysis. The following briefly outlines one such approach:

Analysis – List all identified hazards. Estimate the severity of each identified hazard. Describe the potential causes of each hazard and the probability of each hazard occurring.

Hazard Mitigation – Hazard mitigation occurs in consideration of cost and residual risk. The most desirable mitigation approaches, from the best to the least, are: Design or redesign of the system, the software, or the use of the software (e.g., redundancy or the use of switch covers); protective measures (from the user perspective: passive measures, e.g., safemode); and warning the user (from the user perspective: active measures, e.g., labeling).

Hazard Conclusion – Identify residual hazards after mitigation and determine the level of concern for COTS software; if a *minimal hazard* remains (the possibility of no injury to minor injury), follow the basic requirements, as discussed below. If a *significant hazard* remains (the possibility of death or serious injury), follow both the basic requirements and the special requirements, as discussed below.

COTS Software Verification Activities

Develop Verification Test Protocol – For each test activity, describe the generic approach, procedures, and acceptance criteria for the verification activities.

Then identify all unique (i.e., specific) requirements to be qualified by verification (i.e., those activities that can only be verified by such activities as document and file inspections). For these specific requirements, determine the test requirement, test method, and acceptance criteria for each.

Basic Requirements Identification

Perform the following activities and record the results in the verification report:[2]

❶ **What is it?** Identify the title, vendor, version, date, and such information as the patch number (as applicable).
❷ **What are the computer system requirements for the COTS?** Identify the computer system requirements (both hardware & software).
❸ **What Actions must be taken by the end user?** Identify the end user required or recommended actions (e.g., configuration, data, and changes).
❹ **What does the COTS software do?** Describe the functions, error control, and interfaces (these are the software requirements).
❺ **What does the COTS software *not* do?** Identify the limitations and known problems (bugs).
❻ **How do you know it works?** Using the risk analysis after mitigation, describe the verification and validation activities, including the results.
❼ **How will you keep track of (control) the COTS software?** Describe the training and configuration control of the COTS software.

Special Requirements Identification

Using the conclusions of the hazard analysis, and after the hazard mitigation, if a significant hazard remains, the special requirements must also be fulfilled.[3] These are as follows:

❶ **Product Development Methodologies –** Provide assurance to the FDA that the product development methodologies used by the COTS software developer are appropriate and sufficient for the intended use of the COTS software. (Again, this is stated by the FDA in the context of a medical device, but clearly any use of COTS software with an unmitigated significant hazard should meet these requirements.) The provided assurance should include an audit of the COTS software developer's design and development methodologies used in the construction of the COTS software. This audit must thoroughly assess the development and qualification documentation generated for the software. If such an audit is not possible, and the software represents an unmitigated significant hazard, the use of such COTS software may not be appropriate for the intended application.
❷ **Results of the verification and validation activities –** Demonstrate that the procedures and results of the verification and validation activities performed for the COTS software are appropriate and sufficient for the safety and effectiveness requirements of the system (e.g., the medical device). Verification and validation activities include not only those performed by the COTS software developer, but also those performed by the manufacturer when qualifying the software for use.
❸ **Continued maintenance and support –** Demonstrate the existence of appropriate mechanisms for assuring the continued maintenance and support of the COTS software should support be terminated by the original software developer.

External Data & Transfer Analysis

This low-cost activity is not explicitly addressed by the FDA, but it is recommended for a complete software validation. Analyze all data that is generated outside the software user's team and entered into and used by the software. Verify the data attributes are consistent with the software (e.g., engineering units, formats, and update frequency). Further verify that the method of data transfer is controlled for both the documentation and the media.

Installation Verification

Verify the records of proper COTS software installation. Confirm (and record) that the version and other data identified in the Basic Requirements pertain to the software actually installed. Confirm that the hard-

ware environment meets the minimum of that described. Confirm that the only other installed software is specified (e.g., operating system and other applications). Finally, confirm that the records showing vendor installation recommendations were considered. Once the COTS software is validated to be used under specified conditions and in concert with any other software, a change in that configuration requires a re-validation. For example, if the COTS software is Validated for use with the Windows 95 operating system, it cannot be assumed that it is also validated for use with the Windows NT operating system.

COTS Software Validation Test Activities

Develop validation Protocol – Address each uniquely identified requirement under integration tests and/or system tests.[5] Identify the test method and acceptance criteria for each requirement. Group the previously identified requirements by method (i.e., integration or system-level validation tests) into "single" test cases for the purpose of economy of test execution time. Finally, develop a requirements traceability matrix identifying each requirement and the test in which it is verified or validated.

Structural and Functional Tests – For configurable software, perform structural and functional validation tests as appropriate.[5] There are types of configurable COTS software that may require integration tests, such as the previously discussed analog-to-digital input driver COTS software. In any case, for both configurable and nonconfigurable software, system-level validation tests should be run versus specification and software requirements. Further, it is advisable to provide previously documented experience with the software to support the readiness-for-use of the COTS software.

How Can the Costs of COTS Software Validation be Controlled?

Risk Analysis – Review (or perform) the risk analysis to identify all possible safety problems to the healthcare provider and patient. Determine the severity of the consequences resulting upon failure of the COTS software, and determine the likelihood of occurrence. The determination of depth of testing may be commensurate with the risk analysis.

Criticality Analysis – For software components with a determined risk of minor level of concern and software criticality of negligible to marginal severity, meaning the hazard has an impossible to improbable likelihood of occurrence, determination of depth of testing may be commensurate with the severity and likelihood of occurrence of a failed requirement. For example, one reasonable cost savings may be to perform a visual inspection of the configuration units only, rather than also performing dynamic configuration unit tests. Another means of cost savings may be to perform some validation tests by implicit means. For example, if the output of Module C requires the correct operation of Modules A and B, test Module C explicitly and if Module C performs correctly, assume by implicit means that Modules A and B must also be correct.

Documentation – Consider combining documentation. For example, develop one verification and validation test plan, procedure, and report for appropriately grouped computer systems. These groupings may be for multiple COTS products in one medical device, several COTS products used in different computer systems but all used for the same manufacturing process, or several COTS products used in different computer systems but all used for the same quality system.

What if the COTS Software Vendor Does Not Allow Audits?

What to ask for, what to audit? The principal reason for auditing a vendor is to secure evidence that the vendor is in control of the software development life-cycle activities. These include all areas specified in the FDA's Quality System Regulation,[1] such as proper planning, design reviews, change and configuration management, requirements and design to support updates, and evidence of verification and validation.

What if the vendor does not allow an audit? If the COTS software represents an *unmitigated significant hazard,* the user should consider that the software may not be appropriate for the intended use, and develop the software under the Design Control requirements. If a vendor wishes to allow its software to be used but does not wish to disclose any information that may be considered proprietary, that

vendor may send their Device Master File for the COTS software to the FDA for approval.

Summary

Software is software. The FDA does not declare that Commercial Off-The-Shelf software may be used in any application, critical or otherwise, without being validated just because it is COTS software. COTS software can contain "bugs" and create problems when run in conjunction with other software, just as readily as any other software. However, it is recognized that COTS software has unique aspects to it. By virtue of its potentially wide use, bugs or other limitations or problems may be known and discoverable before use. Another often unique aspect is that COTS software may have many functions that are not exercised in its intended use, whereas software developed specifically for one company's use normally should not contain unused functions (if present, there are reasons to remove them to reduce the complication to the original software development, as well as to the maintenance updates for the software). But the user does not have life-cycle control of COTS software and cannot, therefore, ask that unused functions be removed. However, COTS software need only be validated for its intended use as described in the user's SOP.

COTS software should be analyzed for hazardous conditions just like any other software, and the level of validation should be at least commensurate with the level of risk to the patient or healthcare provider. Beyond that, the company using the software may have business reasons for performing an even more thorough validation of that software to confirm its ability to function properly. ❑

About the Author

George N. Brower has published several articles, and has given presentations at numerous conferences, on his areas of expertise: software development and software validation. He has managed the software development for systems as diverse as a device for the data collection and three-dimensional display of heart sounds to a process control system comprised of seventy-five, dual redundant computers. Brower established and managed the software development and validation efforts for several companies. For the past twelve years, he has served as Deputy Director of the Denver office of Analex Corporation, providing software engineering services for computer systems validation and outsourced software development. Analex-Denver is an ISO 9001 registered company[6] that is the recipient of the 1995 Supplier of the Year Award for software Independent Verification and Validation (IV&V) and the 1996 recipient of the James S. Cogswell Industrial Achievement Award for proprietary and classified data handling. In 1998 Analex received a second award for IV&V testing of critical software. Brower can be reached by phone toll free at 1-888-262-5391, by fax at 303-730-2057, or by e-mail at brower@analex.com.

References

1. FDA Current Good Manufacturing Practice (cGMP) Final Rule; Quality System Regulation, 21 CFR Part 820, effective June 1, 1997, and 21 CFR Part 820.30, Design Control, effective June 1, 1998.
2. FDA Medical Device Software Validation, Guidance for Industry, General Principles of Software Validation, CDRH, June 1, 1997. (Although this reference is a draft document, the principles stated therein are fundamentally accepted.)
3. Guidance for Off-the-Shelf-Software Use in Medical Devices, Draft Document, Office of Device Evaluation, June 4, 1997.
4. Guidance for FDA Reviewers and Industry, Guidance for the Content of Premarket Submissions for Software Contained in Medical Devices, May 29, 1998.
5. Brower, G. N. "Software Structural Testing Methods," *Journal of Validation Technology*, November 1998.
6. International Organization for Standardization, 1994. (Analex is registered for independent software verification and validation, software development, systems analysis, and hardware prototyping.)