

Recipe for Risk Management

Using automated process software in the design, manufacture, and quality control of your device implies that you should validate it to be FDA compliant. This exclusive report shows how this can be done and examines the steps necessary to accomplish it—the analyses, tests, meetings, and documentation.

[1]

This flowchart illustrates one way to plan for increasing validation rigor with more complexity and risk of use.

(click the image to enlarge)

AT A GLANCE

• Explanation of Section 21 CFR 820.70 (i)

• Interpretation of GPSV

• Validation vs. testing

• FDA requirements

If enough risk control measures are put in place so that the residual risk is reduced significantly, the remaining validation tasks can be reduced. David A. Vogel, Ph.D., is president of Intertech Engineering Associates Inc., 249 Vanderbilt Ave., Norwood, MA 02062. An inventor, entrepreneur, and engineer, he founded Intertech in 1982. Intertech provides hardware and software development, requirements, and quality engineering as well as product validation services, training, and consulting for medical device manufacturers. Vogel received a bachelor's degree in electrical engineering from MIT and an MS degree in bioengineering, an MSE in electrical and computer engineering, and a doctorate in biomedical engineering from the University of Michigan. He worked extensively on the development of computer-driven experiment apparatus for vision research at the University of Michigan and on EKG arrhythmia detection algorithms at Becton Dickinson. He is also serving on the AAMI task force on the validation of quality and manufacturing system software. He can be reached at 781-255-5420 or dav@inea.com.

By David A. Vogel, Ph.D.

Manufacturers of medical devices are required by federal law to validate software that is part of the manufacturing or quality system. But there is confusion about what this means. What software does the FDA regulate? What level of validation is required?

There are no checklists available from the FDA. However, an Association for the Advancement of Medical Instrumentation (AAMI) workgroup has been formed to write a technical information report on the topic of software validation for regulated automated process software. This AAMI report will address in detail some of the issues that are discussed here in a simplified form. Meanwhile, there is good

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

documentation that offers guidance.

Regulatory Background

The FDA is good about producing documents that interpret and elaborate on federal regulations it is charged with enforcing. It is useful to look at the regulatory origins to understand what is law and how it differs from the guidance information that the FDA produces to interpret the law.

The federal regulation specifically applying to this software is found in the section on production and process controls. Section 21 CFR 820.70 (i) states: “When computers or automated data processing systems are used as part of the production or the quality system, the manufacturer shall validate computer software for its intended use according to an established protocol. All software changes shall be validated before approval and issuance. These validation activities and results shall be documented.”

The FDA’s January 11, 2002 document titled “General Principles of Software Validation; Final Guidance for Industry and FDA Staff” (GPSV) includes a section that interprets this regulation. Let’s look at how to apply this interpretation.

Is It Regulated?

Medical device manufacturers should inventory all software used in a production, research, engineering, sales, or support facility. That inventory should include the “intended use” for each software item. If the software item is used to monitor or control part of the manufacturing process or part of the quality system, it falls within the regulation.

The quality system includes all software in three general categories.

1. Software used in the design and development of medical devices such as CAD systems, compilers, test software, simulators, and code generators
2. Software used in the creation, maintenance, and control of quality data about medical devices such as complaint-handling systems, lot-tracking systems, training systems, QC systems, and patient-tracking systems
3. Software used in the creation, maintenance, reporting, validation, and storage of electronic records and electronic signatures

Unregulated software can fall under the requirements of 820.70 (i) by its association with other regulated software. A software item that is normally considered unregulated under these guidelines might produce data that is used by regulated software. The validation requirements of 820.70 (i) then apply to that software item. The validation requirements in this case could be focused on the functionality of the formerly unregulated software related to the creation, storage, manipulation, and security of the data used by the regulated software.

Components of Validation

It is a common misconception that validation is synonymous with testing. The GPSV

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

makes it clear that validation includes testing, but validation might also include many other components that lead to the conclusion that the software is fit to use for its intended use. Depending on the intended use, some or all of these components might be appropriate for a given software item.

The adjacent article, "What Validation Includes and How to Handle It," offers advice on dealing with validation.

How Much Is Enough?

The GPSV states: "The level of validation effort should be commensurate with the risk posed by the automated operation. In addition to risk, other factors, such as the complexity of the process software and the degree to which the device manufacturer is dependent upon that automated process to produce a safe and effective device, determine the nature and extent of testing needed as part of the validation effort."

The FDA does not require going further in distinguishing which validation components should be part of a reduced validation effort and which components should be part of a maximum validation effort. The following guidelines are useful as a first pass at determining appropriate validation efforts for regulated automated process software. Every manufacturer's situation is different, and these guidelines should be tailored to the FDA device class and regulatory environment. Part of a quality plan or validation plan for managing automated process software should include the manufacturer's version of the following guidelines.

Guideline #1: Validation Components for Very Low Risk Software Items

Although not specifically dictated by the FDA, certain components of validation should be completed for any validated software item. This minimal subset may be adequate for very low risk software items. Very low risk software items are those whose failure poses a very low risk of affecting the quality of the device being produced or poses a low risk of adversely impacting the integrity or security of quality records required by the FDA.

• The intended use of the software item must be documented to understand dependence of the manufacturer's production or quality system on the software item.

• The risk analysis component must be completed, if for no other reason than to determine how much more validation is required. The GPSV suggests that the level of validation effort is related to the risk posed by the use of the software item. This determination is made in the risk analysis task.

• Configuration management also needs to be considered on each software item. At a bare minimum, the validation report should detail the version of the software item that was validated, the hardware configuration on which it was validated, and the version details of any associated software components that are required for the operation of the software item. Associated software components could include operating systems or PC-based user interfaces for an embedded software item. Concerning spreadsheets, the version of the spreadsheet being

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

validated must be controlled, and the version of the spreadsheet program (e.g., Excel) on which the spreadsheet was validated must also be controlled.

Guideline #2: Validation Components for Medium Risk Software Items

All validation components present for low risk software items are included for medium risk items. Additionally, the following should be addressed.

• Risk management activities should be included. If the risk analysis concluded that there were moderate risks in using the software item, then some type of reduction of the risk should be considered. Risk management can reduce the likelihood of an adverse event related to the use of the software, or can reduce the severity of the adverse event, or can reduce both.

• Traceability should be used to provide objective evidence that risk control measures were implemented and tested for effectiveness.

• Testing components should be added to test the effectiveness of any risk control measures that were implemented as part of the risk management process. High risk functions in the intended use of the software item should also be tested thoroughly to provide some level of assurance that the high risk failures are not likely to occur.

• Management reviews should be added as part of the validation process. They should examine the risk management plans in order to consider the residual risk after risk control measures are in place and to determine if the residual risk is acceptably low to approve deployment of the software item.

Guideline #3: Validation Components for High Risk Software Items

All validation components for low and medium risk software items are included for high risk items. Additionally, the following should be included.

• There should be a detailed life cycle definition for the software, plus a quality plan and verification-and-validation plan detailing quality and validation activities for each phase of the life cycle.

• If the manufacturer develops the software item, requirements should be expanded from the intended-use requirements to formal software requirements. If the software item is purchased from a vendor, then a technical audit should be conducted to assure that good design practices were followed and that vendor-maintained detail requirements are traceable to verification tests.

• Technical evaluations should be conducted throughout the life cycle.

• Testing should be expanded from intended-use testing to full requirements verification. Some attention to unit-level testing and integration-level testing of the critical functions of the software should be considered. If software is acquired from a vendor, audit for test results and known defects.

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

This breakdown of validation efforts commensurate with the risk of using a software item represents only one approach. Manufacturers should use it as a template and make adjustments where necessary. There are many ways to get the job done. They can be defended if the logic behind the plans is consistent and the plans are adequately documented.

ONLINE

For additional information on the products and technologies discussed in this article, see *Medical Design Technology* online at www.mdtmag.com or Intertech Engineering Associates Inc. at www.inea.com.

What Validation Includes and How to Handle It

Below are eight important points involving software and validation.

1. Determination of the software life cycle

The life cycle of a piece of software internally developed by the medical device manufacturer is different from software embedded in a production tool purchased by the manufacturer. Determine what the life cycle will look like to facilitate planning of validation activities at each phase of the life cycle.

2. Document's intended use

The intended use of the software item is similar to a product-level or user-level requirement. Itemize the functions you expect the software to perform.

3. Risk analysis and risk management

Think about how failure of a software item would impact the medical device itself. Do this by considering a failure of each of the functions of your intended-use document. If the impact on the device could adversely affect patient safety, it's an indication that you need to do more to validate this software than you would for one that has a less severe impact. Managing risk includes the identification and implementation of risk control measures. These might include manual or automated checking for failures of the software item. Verification of the output of a software-automated process is a form of validation and in many ways is better than simply testing the software at a single snapshot in time. If enough risk control measures are put in place so that the residual risk is reduced significantly, the remaining validation tasks can be reduced. When dealing with electronic records or signature systems, consider regulatory risks of not having required records available for the FDA and security risks.

4. Configuration management and version control

A configuration management plan should be in place for each software item. Consider who will make the decisions about upgrading the software, who will supply the upgrades, who will install the software, and who will take responsibility for re-validating before it goes online. Consider also any other items in your process that

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

might be incompatible with new versions of the software item and how that incompatibility will be determined and resolved.

5. Planning

Quality plans and software-verification-and-validation plans detail the tasks and deliverables related to an individual software item or the collection of software items that make up an automated process. Plans should cover activities, roles, responsibilities, and resources for each phase of the life cycle.

6. Technical evaluations and management reviews

Determine if the software is technically up to the intended use before putting it online in technical evaluations. Management reviews should reference the output of the tech evaluations and also consider if the residual risk of a failing software item is acceptable prior to deployment of the software. Management also needs to consider whether the users of the software are trained sufficiently to successfully deploy the software. The management review should be considered the final gate to deployment.

7. Testing

Focus testing on existing risk control measures and then on functionality of the software whose failure could lead to severe consequences. There is little value in testing functions of the software that have little effect on the manufactured device. For electronic record systems, focus testing on functions whose failure could result in record loss, alteration, or loss of security.

8. Traceability

You'll need some documented trace to show that all the functionality detailed in the intended-use document was implemented or acquired. Furthermore, any risk control measures identified in your risk management report must be traced to where it was implemented and where its ability to mitigate the risk was tested. Finally, functional testing of the software should be traceable back to software requirements or intended-use functionality.

Top 10 Questions to Ask When Choosing an Outsource Vendor

Outsourcing can be better than in-house development, but you have to choose your outsource partner carefully. You'll want to be sure that the people on the vendor team worked together before and that they'll still be there two years from now when it's time for an upgrade. You'll also want to avoid a vendor that is developing products that might compete with your products.

Medical device manufacturers often consider outsourcing the development or validation of their embedded-device software or their manufacturing/quality system software. Many manufacturers believe a specialized outsourcer can do the job faster, cheaper, and better and can provide an experienced team that will reduce the risk of out-of-control projects.

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

Here are 10 questions to ask a prospective medical device outsource partner to make sure you will achieve your objectives.

1. How much experience with medical devices or medical device companies do you have?

Be skeptical if the vendor acts as if all development is alike. This is a regulated industry. The FDA has specific expectations for design controls and validation.

2. How do you control the quality of your design, development, or validation output?

Ask to see the vendor's standard operating procedures and examples of work done on a similar project.

3. If you develop software for me, will I exclusively own the rights to all components of the delivered software?

Many contract developers re-use low-level software functions from client to client. This means you may be paying for part of the development of the next client's project. Even worse, you may end up being sued if the software developed for you contains components developed at another client's expense.

4. How many engineers working on my project are full-time employees and how many are contractors brought in just for my project?

Too many independent contractors can lead to an unfocused team with competing priorities. The likelihood of getting the same team for future updates is very low if independent contractors are used.

5. How will the project be managed?

Is the outsource vendor expecting you to manage its engineers or is it providing project management? If the vendor is providing management, what does it cost and how will you pay for it?

6. What are the deliverables you will provide me as the project progresses?

Know what to expect for your money. Find out if you will be able to approve intermediate results before the outsource partner moves on to the next phase. This can save both time and money.

7. Who will pay for mistakes?

No sizeable project can be done error-free. Who is responsible for the costs of fixing mistakes? (Gross negligence and incompetence are other issues.) Certainly, the vendor won't do it for free. Will the costs for repairs be done on a time-and-materials basis, budgeted in contingency line items, or covered by markups in the

Recipe for Risk Management

Published on Medical Design Technology (<http://www.mdtmag.com>)

rates or by additional fixed price increments? You need to know how it is handled so that you don't pay twice.

8. Is this a time-and-materials contract or a fixed-price contract?

If it is a time-and-materials contract, what information will you be given to make sure your money is being spent appropriately? If it is a fixed-price contract, how will out-of-scope tasks be handled? What if the task can't be done for the fixed price? Is the vendor large enough to take large losses?

9. How do you handle confidentiality with your employees and contractors?

Remember that some of the engineers on your project may be working on a competitor's project next month. What assurances do you have that there won't be any crossover of intellectual property or leakage of trade secrets? Has the vendor worked for a competitor in the past?

10. What is the best project you have done?

Find out what the best work has been as well as what went wrong on a project that didn't go so well. And then, work with the vendor to make your project look like the most successful project.

Source URL (retrieved on 03/08/2014 - 7:29am):

<http://www.mdtmag.com/product-releases/2005/02/recipe-risk-management>

Links:

[1] http://www.mdtmag.com/ProductImages/0502/md52COS_lrg.jpg