

Ideas for Validating Production Software

High Risk Production Software Doesn't Always Need an Elaborate Testing Program

by *David A. Vogel, Ph.D.*
Intertech Engineering Associates, Inc.

and *Kevin J. Barnes*
Apogee Designs, Ltd.

as published in *Medical Design*, September 2005.



This article outlines a risk-management technique that prioritizes software validation efforts so resources are spent where they add value to production and satisfy the regulatory intent. Automated inspection equipment can ensure the quality and reliability of the manufactured device and contribute to the validation of other production software.

The FDA requires medical-device manufacturers to validate software that is part of production or a quality system. But what does that mean? Those charged with the job are often not software engineers nor do they have backgrounds in validation or regulatory issues. As a result, some manufacturers fall short of validating quality-system software for its intended use.

Production software is part of the “quality system.” It’s not embedded in a medical device. It’s software in production tools, for process monitoring, and control applications.

The Food and Drug Administration (FDA) suggests the validation rigor relate to the amount of risk the software poses. This doesn’t necessarily mean subjecting high-risk software to elaborate and expensive testing programs.

Regulatory background

The federal regulation that applies to quality-system software is found in 21 CFR 820.70 (i) on Production and Process Controls. It reads: “(i) Automated processes. When computers or automated data processing systems are used as part of 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.”

COMPANY PROFILE

Intertech Engineering Associates, Inc.

Address: 100 Lowder Brook Avenue
Suite 2500
Westwood, MA 02090
www.inea.com - (781) 801-1100

Industry: (Electro)Medical Devices

Services: Assessments
Training
Consulting
Hands-on Engineering

Skills: Product Design
Risk Management
Requirements Engineering
Electronics Development
Software Development
Software Verification and Validation
Production/Quality System Software Validation

Section 6 of the FDA’s document, General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002 (commonly referred to as GPSV), gives some advice for applying validation techniques to production and quality-system software.

Of course 21 CFR 820.70 (i) covers more than production software. Medical device manufacturers should inventory all software that automates any aspect of medical-device production. The GPSV, for instance, mentions programmable logic controllers, digital function controllers, statistical process control, supervisory control and data acquisition, robotics, human-machine interfaces, input/output devices, and computer operating systems.

Validation is more than testing

Validation is not synonymous with testing. But the GPSV makes it clear that validation includes testing and might also include actions that lead to the conclusion that the software is “fit for its intended use.” Some or

like to facilitate validation activities at each phase of the lifecycle. The accompanying waterfall charts provide two examples.

Document intended use

For example, itemize each function or command. It is important to document each item in an Intended Use document along with how it influences the production process. Understanding how each command or function impacts the overall production process, prioritized by risk, lets users take advantage of the following process-level validation techniques.

For example, consider a Visual Basic program written by an ambitious production engineer that controls a grinding machine for sharpening cutting edges onto surgical scalpels. Assume the software has three commands: one to select the grinding profile (the shape of the blade), one to select the grinding pitch (how fast the edge tapers), and one to initiate a grinding sequence on a scalpel blank. A failure of any of these functions will have a serious effect on the finished product. The intended use of the software



all these components might be appropriate for a given program, depending on the intended use and associated risks. Validation includes:

Determining the software lifecycle.

The lifecycle of software developed in-house by a medical-device manufacturer differs from software embedded in a production tool purchased by the manufacturer. Determine what the lifecycle will look



is to control the grinding machine to sharpen scalpel blanks to an accuracy of x degrees of specified pitch, and y mils of specified shape. The pitch-set command takes a numeric entry from the machine operator to define the scalpel-edge pitch ranging from a to b degrees in increments of c degrees. The profile-select command selects from files on disk that define the shape on a grid of resolution d mils.

Manufacturing a low-cost line of scalpels this way sends them from grinder, to sterilizer, to packaging without inspection. Production greatly depends on the proper operation of the grinder. A second, higher-cost line of scalpels manufactured by the grinding machine subjects the scalpels to an individual inspection to assure the proper profile and pitch. Because each scalpel is inspected for defects and eliminated if found, a failure of the software will have less impact on the production process.

Risk analysis and management

Consider how the failure of a software item would impact the medical device in production. Do this by considering a failure of each of the functions listed in the Intended Use document. If the impact on the device could adversely affect patient safety, it's an indication you must do more to validate software than you would to validate a command that has a less severe impact.

Continuing with the scalpel-grinding example, the risk of failure for the low-cost scalpels not inspected after production is high relative to the higher-cost inspected scalpels. We have not defined any other risk control measures for the low-cost line. Therefore, we depend more on the correct operation of the software controlled grinding machine.

Managing risk includes identifying and implementing risk-control measures. These might include manual or automated checking for failures of the software item. Verifying the output of a software-governed process is a form of validation, and is preferable to simply testing the software once. Putting enough risk-control measures in place to significantly reduce residual risk can reduce the reliance on remaining validation tasks.

Knowing when enough risk control measures are in place is a matter of understanding what level of risk is acceptable to your end users (and to your business), and being able to quantify the residual risk. Since risk is a combination of severity and probability, one can reduce risk by reducing either the severity or the probability of failure. Unfortunately, the probability of software failure is difficult if not impossible to quantify. Therefore, it is better, where possible, to focus efforts on reducing the severity of a failure, or to reduce the process probability

of failure. We can tolerate software errors by detecting and correcting them downstream in the manufacturing process.

This is key to using automation to reduce software-testing activities. Verifying 100% of the output of a software-controlled process validates the correct behavior of the software. For this logic to work, the failure modes that result in a severe failure must have corresponding verification tests on the finished product.

Configuration management and version control -

A configuration-management plan should be in place for all software. This plan should identify who is responsible for decisions to upgrade the software, who supplies the upgrades, who installs the software, and who takes responsibility for re-validating before it goes online. Consider also other items in your process that might be incompatible with new versions of the software, and how that incompatibility will be determined and resolved. The plan should also require a new risk assessment at the installation of each software upgrade. New capability in upgraded software could result in new usage modes that would require validation.

Planning

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

Technical evaluations and management reviews

In technical evaluations, determine whether or not the software is technically up to the intended use before putting it online. Management reviews should examine the technical evaluation records and consider whether the residual risk of failure is acceptable before the software is deployed.

Consider this imaginary management review. It's conducted for a piece of software critical to the operation of a production line. This is management's opportunity to ensure that everything possible has been considered in making sure the software will operate



systems, focus testing on functions whose failure could result in record loss, alteration, or loss of security.

Traceability

Some documented tracking is needed to show that the capability detailed in the intended-use document was implemented or acquired. Further, any risk-control measure identified in the risk-management report must be traced to where it was implemented and where its ability to mitigate the risk was tested. Finally, functional software tests should be traceable back to software requirements or intended use capability.

Let the validation effort reflect risk

The GPSV says, “The level of validation effort should be commensurate with the risk posed

The technician is using equipment from Apogee Designs Ltd to verify 100% of the outputs from production software on the line. The FDA says manufacturing software that imposes a high level of risk on a final device must also be evaluated and validated before it is put to work on production lines.

properly when deployed. Managers may note that several versions have been tested with little decline in the defect rate. Production engineers have noted in evaluations that the new releases of software are being produced at a rapid rate and have concerns about their stability. Several high-severity errors were detected on the last few versions. This is a case where management should be concerned that severity of failure has not been controlled, and that the probability of failure is high because the software has not matured as evidenced by the rapid release rate.

Management should also consider whether the software’s users are sufficiently trained to successfully deploy the software. The management review could be considered the final decision point to deployment.

Testing

Focus testing on risk-control measures in place, then on failures that could lead to severe consequences. There is little value in testing software functions that have little effect on a manufactured device. For electronic-record

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.” That’s a long way of saying the greater the risk of operation, the more you must test to assure that failure will not occur.

Unfortunately, the FDA does not further distinguish what validation components can be part of a reduced validation effort, and those that should be part of a maximum validation effort.

It’s beyond the scope of this article to detail how a validation effort can be scaled so it is commensurate with risk. It’s usually hard to justify skimping on the documentation of intended use, risk assessment and risk management, configuration management, planning, and management review activities. The thought, discussion,

and debate required by these activities may well lead a validation team to the conclusion that a software item is of low risk. In fact, these are the activities that are necessary to come to that conclusion, and that is why it would be difficult to reduce or eliminate them. The main activity that can be scaled to risk is the testing effort.

How to test

Not surprisingly, there are different ways to test different types of software. Software developed in-house, with full knowledge of the specifications, design, and implementation, could be tested at a unit, integration, and system level much like device software. Off-the-shelf software may come with few specifications, design documents, or source code. Some medical-device manufacturers have embarked on elaborate and expensive exercises to reverse-engineer requirements, specifications, and designs to test against. In some cases, the software



Apogee's automated testing device would show users which gauge to use for the test at hand.

vendor is “audited” by the medical-device manufacturer in an attempt to document that the software was designed and developed in a controlled environment. The rationale is that process-driven development is less subject to error than chaotic development.

While satisfying regulatory intent, these activities do little to “prove” that the software is safe for its intended use. Reverse engineering probably guesses much less

than 50% of the capability of a piece of software, and even less about design and implementation. Although we advocate process-driven development, defects occasionally slip through. The bottom line is that “validation activities” for off-the-shelf software are time-consuming and expensive, weak at best, and do little to add assurance about the software’s safe operation in the production process.

Unfortunately, much of the software used in a production process *is* off-the-shelf, or embedded in production machine tools. The software can control critical elements of production. If the software fails, it could be disastrous for the product and, ultimately, the product’s end user. So, how can you assure the proper functioning of the software?

An alternative to testing software once against a set of real or assumed requirements is to test its outputs for each medical part it is responsible for creating. This is what we referred to as 100% verification of outputs. It is not testing to provide a level of assurance that the software is error-free. It is testing each output to detect and correct software malfunctions. This type of verification adds value to the production process and can be automated. Here’s how.

Medical device manufacturers can benefit from the experience of other high-reliability, regulated industries such as automotive, defense, and aerospace. Military protocols for the assembly of mission-critical devices are as demanding as those required under FDA’s 21 CFR validation requirements. Errors cannot be tolerated when building miniaturized, high-tolerance, low-failure devices, such as those used in guidance or life-support systems. Assembly processes must be monitored and periodic quality evaluations recorded at various steps.

Automated Quality Testing and Automation

An industrial design group, Apogee Designs, Ltd., Baltimore, (apogeedesigns.com) has created an interactive workstation that provides real-time validation of all assembly operations and verifies embedded software operations. Such systems can also guide and verify manual operations associated with production, thus improving the quality and reliability of the device.

The base technology consists of a chassis with custom I/O boards and peripherals that create an interactive workstation. Components can be mixed and matched, or customized to satisfy quality, assembly, or training requirements.

The Apogee concept originated with workstations developed for a transmission manufacturer to simplify testing and collecting of critical quality data. The system constantly reviews and documents production quality, using state-of-the-art digital gauges at statistically derived times and points in the assembly operation. Complex items require up to 60 different measurements and use many different instruments.

Workstations have a mounting platen, a tool, an instrument storage array, and a video monitor. Tools, parts, and quality instruments mount on the workstation, each with an identifying “use me” LED. The mounting platen holds the assembly and positions it for easy access to areas being assembled and tested.

Each step of the procedure is presented on a display and the appropriate LED identifies the instrument or tool required at that point in the assembly/test process. Video directions do not advance unless the active event is executed and properly validated. Aside from functional validation, video recognition systems can be incorporated to confirm full installation compliance, such as each screw is in place and installed with appropriate torque. The system displays real-time Go/No-Go or numeric data at the click of a mouse. When in

range, it records the data on the next click. All relevant information, along with video documentation, can be downloaded to track short and long-term trends.

When properly instrumented, such a system could be used for final inspection, measurement, and test for all critical specifications that are under software control, thus verifying 100% of the software output. Of course, the system need not be dedicated to a single manufactured device. It can be programmed for verifying any number of manufactured parts.

Additional plusses include providing an ergonomic and efficient workspace. Images and work instructions combined with pick-to-light indicators on tools and parts bins greatly reduce the likelihood of incorrectly performing a manual operation. Images and procedures are easily configured with basic spreadsheets. The systems provide personnel flexibility through cross-training and work standardization with constant visual feedback.

Personnel are visually guided in the step-by-step process, automatically recording each test result. The sequential process precludes operator changes in specified tasks, and documents each task along the way. In the case of the transmission manufacturer, training for a technical assembly and inspection process was cut from over two weeks to just a day.

Of course, systems like this are driven by software and subject to 21 CFR 820.70(i). They may also be subject to Part 11 regulation for the electronic records the system maintains. However, the automated inspection and test system may substitute forelaborate testing of any number of software-driven tools and upstream production operations. Suppliers of the system may deliver end users with their own validation and testing data, which can further reduce the validation load on medical-device manufacturers.

ABOUT THE AUTHORS:



David Vogel is the founder and president of Intertech Engineering Associates, Inc.

Dr. Vogel was a participant in a joint AAMI/FDA workgroup to develop a standard for Critical Device Software Validation which was subsequently included in the IEC 62304 Software Lifecycle Standard. He was also a participant on the joint AAMI/FDA workgroup to develop a Technical Information Report (TIR) for Medical Device Software Risk Management. Currently, Dr. Vogel is a member of the AAMI/FDA workgroup developing a TIR on Quality System Software Validation.

A frequent lecturer for workshops and seminars on topics related to medical device development and validation, Dr. Vogel also is the author of numerous publications and holds several patents.

Dr. Vogel received a bachelor's degree in electrical engineering from Massachusetts Institute of Technology. He earned a master's degree in biomedical engineering, a master's degree in electrical and computer engineering, and a doctorate in biomedical engineering from the University of Michigan.

Kevin Barnes is a vice president of Apogee Designs Ltd., 101 South Kane Street, Baltimore, MD 21224. Apogee Designs is an industrial and engineering design firm that specializes in the design, development and assembly of mission critical equipment including medical devices. Kevin can be contacted at kjb@apogeedesigns.com or (410)633-6336 Ext. 28

ABOUT INTERTECH:

Intertech Engineering Associates has been helping medical device manufacturers bring their products to market since 1982. Through a distinct top-down service model, Intertech offers high-level consulting and hands-on engineering. By balancing technical expertise and practical business experience, we support clients through all phases of product development. While we do make your job easier, Intertech exists not to replace but to partner with clients to help balance the concerns of quality, time and cost.

With considerable experience in FDA regulatory compliance, our time-tested development process can anticipate and solve problems inexpensively on the planning board rather than through costly solutions later in the development, test, or post-deployment phases. By using deliberate processes, Intertech ensures an improvement in quality and can build client expertise.

Call us today for more information or a free consultation at 781.801.1100

Intertech Service Offerings:

*Risk Analysis and Management
Software Design and Development
Electronic Design and Development
Requirements Development and Management
Documentation and Traceability
Verification and Validation
Evaluations, Reviews, Inspections
Planning
Project Management
Compliance Consulting and Training
Manufacturing and Quality System Software Validation*

INTERTECH Engineering Associates, Inc.

100 Lowder Brook Drive Suite 2500 Westwood, MA 02090 USA

www.inea.com - info@inea.com - Tel: (781) 801-1100 - Fax: (781) 801-1108