

Features

## Validating Your QMS the 'EZ Way'

Securing approval for an eQMS solution is only the beginning; ensuring it's properly installed and set up is arguably more significant.



James Shore and Roxana Rosales 04.01.20

Congratulations, you were able to get approval for the electronic quality management system (eQMS) you've been dreaming about, which is going to take your quality system to the next level. Now the fun begins—validating the software. This article is intended to walk you through the basics of what to do next as well as what to watch out for and what to avoid.

### Why Validate?

Following are several specific requirements for reference in case your management asks why validation activities are necessary. From a regulatory requirement, the top four cited are:

- <sup>1</sup> General Principles of Software Validation; Final Guidance for Industry and FDA Staff, Document issued on Jan. 11, 2002: "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."  
FDA 21 CFR 820 Quality System Regulations - 21 CFR Part 820.70(i): "...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."
3. ISO 13485:2016(E) Section 4.1.6 establishes: "The organization shall document procedures for the validation of the application of computer software used in the quality management system. Such software applications shall be validated prior to initial use and, as appropriate, after changes to such software or its application. The specific approach and activities associated with software validation and revalidation shall be proportionate to the risk associated with the use of the software."
4. 21 CFR Part 11: "In the event that these systems have associated electronic records or signatures, they are subject to the Electronic Records; Electronic Signatures regulation."

If attention was focused on more than just the need to satisfy regulations, other benefits of validating the QMS can be realized. These include:

- Data integrity, which is becoming a hot topic in the medical device manufacturing industry
- The creation of optimized processes that reduce cycle time and increase efficiencies
- Easier to search one location to find information quicker

While these benefits are excellent for the company, they are not the critical factor with regard to a QMS. Rather, the major challenge is convincing those at medical device companies who are reluctant to invest in the automation and technology to help more effectively run the business. The majority of these companies still rely on paper or general-purpose tools for their documentation and quality management systems.

## Validating Your Software the 'EZ Way'

The "EZ way" begins with a clear understanding of the actual process by mapping it out. The second step is to eliminate the activities that don't add value. Finally, the process is automated. This is essentially the Lean way of thinking: Never automate a process you don't know well and can't do well manually first. We have worked with many customers who have not performed these initial steps and, ultimately, did a great job automating an inefficient process.

### The Nine 'EZ Way' Steps

- Process Map: Understand the process
- Master Validation Plan: Put together the objective and define success
- Risk Assessment: Risk identification and mitigation
- Test Plan and Resources: How it will be tested and who is needed
- Installation Qualification: Setting up
- Operational Qualification: Verifying the installation
- Training: Show how to use the software
- Performance Qualification: Testing the actual process
- Change Management: Keeping the validated state under control

When reviewing the activities, it becomes apparent the process is not too different from the process validation performed within a facility's production.

**Process Map**—Before starting the software validation, ensure the actual process is documented. Using a flowchart to identify the steps, who is performing them, and what the decisions are is recommended. This allows you to determine how to customize the software to specific processes. In some cases, the software may not allow customization of specific steps so the flowchart enables you to identify the gaps. In that situation, the process needs to be modified or activities outside of the software need to be created. It is not uncommon for specific steps to not be part of the software or the customization to not be possible. For example, the impact analysis of the inventory when making a change isn't part of the system so that report may need to be provided separately and attached to the document change request.

**The Plan**—Once that is accomplished, the next task is the creation of the Master Validation Plan (MVP). The MVP outlines what software/version is being used, what modules or processes will be implemented, and what they are supposed to do [e.g., change management, corrective and preventative action (commonly referred to as the intended use)]. Most eQMS systems are module-based, so specify which modules will be implemented. Just like a process validation, the plan must include how success will be defined and how to deal with failures/deviations. The scope of the plan needs to be defined so the risk assessment can be properly determined.

**Risk**—The risk assessment needs to review the software, what activities will be done, and the level of concern (LOV) if failure occurs, as well as the impact to the patient, business, or compliance. The ratings are usually identified as Major, Moderate, or Minor, which need to be defined by the company. The importance of identifying the LOV is to determine the level of validation and testing required. For example, the final inspection measurements are output to a report that goes with the device. The physician needs to know these dimensions to perform the procedure correctly. In this case, the risk would be evaluated and the LOV would be Major so the validation activities regarding how the data is entered, verified, and reported would be in depth.

**Resources**—The amount of work and resources can be identified before starting and can be added to resource requests. It's critical you have the right people available when you start the testing.

**Installation Qualification**—As part of the selection of the software solution, it is important to know what software validation packages and support are offered by the provider. Most providers offer the software validation packages as part of their offering. The documentation will cover the Installation (IQ), Operational (OQ), and Performance (PQ) steps and records of the tests.

Depending upon where the software will be hosted, the IQ will cover the installation of the software on the server or how it will be provided virtually (e.g., cloud-based software). This plan needs to capture the steps of the installation, who performed the activity, and the results showing it was accomplished correctly. Though this is usually the easiest step, there has been many cases where the installation was not performed properly due to a software corruption issue or the wrong software revision was installed. Once

that is completed, a report needs to be written and approved before going to the next step. Though it may sound odd for pointed out, there are plenty of audit findings documented because these reports were not created.

**Operational Qualification**—These activities verify the software was both installed and operating correctly. The essential tests can be accomplished by logging into the software, going into each module, and ensuring the process steps work as specified by the software provider. Most providers will supply the steps and data that show the activities that need to be performed. These steps need to be defined before testing occurs and the person performing the steps needs to document the activities and indicate whether they passed. If they did not work correctly, the issue should be documented and an investigation performed. All of these issues should be resolved before performing the PQ tests or it will jeopardize the entire validation activity. Again, the report should be written and approved before going to the next step.

Depending on the software solution, the configuration of the software may have to be setup during the OQ or PQ steps. At this stage, users are entered into the system, security and access for certain modules and activities are established, and data and files are uploaded or imported into the database. There should be a process to validate the information was properly set and the data was uploaded correctly. Data integrity is critical and it would be a major failure if the data is not correctly entered. Most software solutions, however, have easy-to-use tools to accomplish these tasks. Once more, ensure the results are written and approved before proceeding. Remember, if it wasn't documented, it didn't happen.

**Training**—Before beginning the next steps, this is usually a good time to conduct the training of the employees. The easiest way to train users effectively is to use instructions with screen shots and show them how it works on the system. In some cases, off-line test environments can be created where the users can go in and get accustomed to the software. After they've become acclimated, creating some simple tasks to get them comfortable using the software is recommended. Please understand the level of adoption and efficiency will be directly proportionate to the level of support provided in the beginning. Change is not always easy for everyone and this step is critical to success.

**Performance Qualification**—These activities should be the easiest part as it involves simply following the flowchart but only in the electronic process. For these tests, use examples of items that have already been processed in the current process. A good example is to take the release of a new quality procedure that needs to be approved. The test plan should document the steps that need to be performed and each should be recorded as the user conducts the test. This is commonly referred to as "use cases" in the software world.

A best practice would be to get the actual people involved in the process into one conference room with their computers, where they would log into the system with their user name and password. For example, the process could be started by the originator initiating the document change request. The documentation control person receives notification of the request, reviews it, and pushes it into the review process. The approvers accept the change and the document is released. Also, the process should be tested for all scenarios such as if the initial submission is rejected by documentation control or not all the approvers accept the change. The process flowchart should be well defined to show what happens next and the software should perform as intended. If the software doesn't work as intended per the test plan, the test should stop and the investigation to determine the issues that need to be resolved before going further started. Changes to the software may need to be made and the test restarted from the beginning. The pain experienced at that moment will be much less than if this happened after the software was in production use.

After completing all the use cases and addressing any issues, the report needs to be written and approved stating the software has met the intended use and is approved to use in production. Similar to any process validation, the report needs to clearly state the results are acceptable and the risk of any known issues have been mitigated.

**Change Control**—While the software is now running smoothly, the task is not completed. Like any other validated process, there is a need for controls to monitor the validated state. In the beginning, monitor the process closely and give feedback to the specific people experiencing issues. You may need to revisit the documented procedures to see if an update is required. In addition, start a log of any changes made to the configuration/settings and any software releases that come from the software provider. In both cases, document the impact of the change and whether or not validation is required. If not, ensure the rationale is clear as to why it was not required (e.g., software patch of a module not used). Also, after so many changes are made, perform a partial validation using the use cases used to perform the initial validation. Those tests should be documented and approved, as well as clearly documented if the validated state is still in control or further activities have to be completed.

## Things to Watch for and Avoid

There are a number of considerations to keep in mind when implementing the eQMS. Following are several of which to be mindful.

Before starting the software implementation, create a matrix to determine permission levels or privileges for each role. Every person should fit into a role that allows certain access to specific tasks within the system. There have been plenty of audit findings where the matrix is not properly documented, and users had unlimited access to perform transactions for the final QC release.

Document the system version, the platform specifications of the host server, and the version of Windows the users have on their computers. A software upgrade, such as to Windows 10, can cause a version of the eQMS not to work correctly. With no contingency plan documented, this can cause a major shutdown in production.

Begin with a pilot or limited scope of the eQMS since the change to using software may have some impact on the business. Regardless of how well you plan, test, and train, something will emerge that will create an issue. This approach allows you to learn not only how the software really works but also how the people involved use the software. During this time, run the current process and software process in parallel for contingency reasons. It may be extra work but it will be better than having people lose faith in

the software.

Don't perform use-case testing logged in at admin-level permissions. Essentially, the reason to test with actual users is logged in with the permissions assigned to their roles. This ensures the functions work "as intended" and prevents use performing certain tasks to which they don't have access. Software validations can be cited for not testing the permis

Once the software is validated, evaluate all changes the software supplier releases before implementing them. Create a simple log to keep track of the versions/revisions, identify the change, and document if the change impacts the validated state. If there isn't any impact, document the rationale. Always remember, if it wasn't documented, it didn't happen.

Set a frequency to evaluate the validated state and determine if any validation needs to be retested. After so many changes have been made to software, determining if the system is still running as originally tested is a good practice. Just like any validated process, there need to be checks to ensure you are still in control. Use a risk-based approach to identify certain elements that could be retested against the original validation plan. The record can be used to show the process is still validated. Remember, this isn't a "one and done" activity; validation is a dynamic process.

## Conclusion

The use of eQMS software will greatly improve the efficiencies of good processes and will reduce the amount of time lost looking for the documents. This is not one of those solutions, however, that allow you to install and run by the next day. The implementation approach is very similar to process validation activities and requires some slight adjustments on the verbiage used. There are plenty of user groups online for the software, which can serve as an excellent resource for questions, to see how others have resolved issues, and discover best practices. The validated state needs to be monitored with changes that happen, but similar to a fine-tuned car, it will perform as well as you take care of it.

---

**James Shore** is the chief quality officer at Quality Lean Solutions ([www.qualityleansolutions.com](http://www.qualityleansolutions.com)), which provides "Simple Sustainable Solutions" for medical device and high-tech manufacturers. He has over 30 years of quality and supplier management experience working in the medical devices, semiconductor, aerospace, and defense industries. Shore's professional certifications include ASQ Certified Six Sigma Black Belt, ASQ Certified Quality Manager/Operations Excellence, Certified Quality Auditor, and Certified Mechanical Inspector, as well as ASQ Senior Member (joined in 1988). He is a Certified Welding Inspector and Test Supervisor for the American Welding Society as well.

**Roxana Rosales** is a medical device professional who works with Smith+Nephew as senior quality control manager to lead their quality control department. She began her career at Hospira Costa Rica in 2004, and worked for ICU Medical after Hospira's spin-off as strategic maintenance engineering manager, leading the implementation of a predictive maintenance program. After spending over 15 years working in multimillion dollar medical companies, in different positions such as electronic system integration engineer, software quality validation engineer, calibration supervisor, equipment engineering manager, and site validation engineering manager, Rosales knows what truly drives regulatory compliance from operations and quality perspective. She is a Certified Lead Auditor in ISO 13485 with experience in automation, calibrations, auditing, software validation, and process validation. She can be contacted by phone at +(506)6040-1182, email at [rosalesb.roxana@gmail.com](mailto:rosalesb.roxana@gmail.com), or at [www.linkedin.com/in/roxana-rosales-briceño](http://www.linkedin.com/in/roxana-rosales-briceño).

---