

# Seven Keys to Successful Software Validation

Natalie J. Kennel RAC, ASQ CQE & CQMgr.  
Principal, NJK & Associates, Inc.

## Background

Companies purchase computerized business systems to keep track of everything from the purchase of components to their manufacturing processes. These systems often require customization in order to fully meet the business needs of the organization, and ongoing support as the business grows and changes.

In FDA regulated industries there is a requirement that all software used for any quality system task be validated prior to its use, and there are additional business reasons why validation makes sense even if it were not mandated.

The entire process of choosing the right software, deciding upon its customization, and validation of the software to meet regulatory and business requirements requires a significant investment of time and resources.

What follows are some highlights of what I've learned over the years that can help any business to avoid surprises and have a successful validation and implementation.

## Does This Apply to My Company?

- It does if you are designing, developing, manufacturing, or distributing a product that is regulated by the FDA, or if you are conducting clinical trials.
- It does if you are using or plan to use the software for any GxP function.
- It does if your company is publicly held.

The FDA requires that business software be validated because in their words, “software is different”. Unlike manufacturing processes whereby errors are of a statistical nature, software bugs are deterministic – and can have profound impact on the validity of the information obtained and/or actions taken.

It is not only good practice to ensure that your business software works as intended, but a SOX requirement for publicly held companies.

## The Keys

### ***1. Obtain Management Support***

Quality, Regulatory, and Operations management should already understand the need for validation. Senior management may not. It is important that all involved understand that undertaking this project will require money, the time of company employees, and other resources. **Without management support at the outset you will expend more and achieve less.**

## ***2. Build and Educate Your Team***

But first, educate yourself! Business software validation requires a team effort as the different functions within your organization have differing needs and can contribute in various ways. Plan to include at least Information Technology, Quality Assurance, Regulatory, Clinical, Operations, Finance, as well as outside resources.

Learn the lingo and the regulatory requirements. Even if you plan to use an outside consultant you are ultimately responsible for asking the right questions and ensuring that the plan is properly executed.

Become acquainted with the terminology: Software validation, design verification, process validation, design validation, intended use validation, COTS, black box testing and white box testing, regression testing, and Part 11.

## ***3. Determine the Intended Use and Requirements for the Software***

In this step you document what you want the software to do, and what functions and activities it will support. Here you will explicitly record the requirements to achieve these tasks as well as the regulatory requirements. You will use this information to validate the software for your intended use, and for the evaluation of the software package itself, the customization necessary and any configuration choices. Performance of these tasks prior to purchasing the software package can help to avoid costly mistakes.

## ***4. Assess Risks***

Here is where you consider product safety, regulatory compliance, and business risks. You ask yourself questions such as: “What can go wrong if the software doesn’t work properly?” and “What can be done to eliminate or reduce these risks?”.

## ***5. Determine Related Company Readiness***

Your IT security infrastructure must require unique user names and passwords, enforce regular changing of passwords, support user groups, and contain other controls. There must be a back-up method established and processes for testing and disaster recovery. This is also the time to write an SOP on Software Validation.

If the software will be used in manufacturing you will need to establish a part numbering system, policies and procedures on part numbering, a change control procedure, and the interrelationships between Regulatory and Finance. The business processes need to be established in order for the software to support them.

## **6. Generate a Plan & Execute it**

Next you will develop a software validation plan based on the requirements and your risk analysis. Test cases are written and approved. The acceptance criteria must be documented and approved prior to executing the test cases. If any of the test cases fail you'll need to investigate and resolve the issues and may need to repeat the test cases. After they all pass and the report is written and approved you will need to formally release the software for use.

The test report must contain or reference the results of all of the test cases, as well as identification of the software under test, its version, and the hardware it was tested on. It will also include who did the testing and a trace matrix of each test case to the original requirements.

## **7. Plan for Changes**

There will be changes. Establish a method of configuration control – this works well if you use the process you already have established for configuration control of your controlled documents and records.

Determine what will trigger revalidation of the software, and how such testing will take place. Remember that this should always be risk-based.

## **Summary**

- You don't need to be a software expert to have a successful validation.
- You do need to be a knowledgeable consumer.
- Determine your company's requirements and intended use for the software.
- The degree/extent of validation should be risk based.
- Develop test cases based on the requirements and on intended use.
- Conduct test cases and use tracking matrix to ensure that all requirements are covered.
- Resolve any issues discovered.
- Document the final results.
- Revalidate when changes are made.
- Focus on the predicate rules.

## **Conclusion**

By following the steps described above you will add value to your company, and will help to avoid unanticipated and unwelcome consequences.

You win when the software you implement meets your intended use and user needs.