ProcessPro

10 Steps to Software Validation:

Creating a Plan to Save Time and Money

Software validation confirms that certain specifications coincide with user needs, the software is meeting intended use and requires objective evidence that the requirements can be consistently fulfilled. This is required for any company covered by the Food, Drug and Cosmetic Act and 21 CFR Parts 210 and 211. Also, if a company is keeping current Good Manufacturing Practice (cGMP) data electronically and relying on that information to make cGMP decisions, they are required to perform software validation. Manufacturers who continue to see increased enforcement of these regulations include pharmaceutical, nutritional supplement and cosmetic companies.

For small to mid-sized manufacturing companies, software validation can seem like an overwhelming task. However, the benefits of validating prove to be well worth the cost. Some of the reasons for software validation include:

- Saving money by discovering weaknesses or flaws in processes prior to production
- Providing management with a clear understanding of risks so that appropriate decisions can be made
- Increasing the likelihood projects will be completed on time and within budget
- Increasing product and system quality
- Meeting regulatory compliance

Creating a software validation plan before commencement will save you time and money and make the undertaking manageable. It’s important to find a validation team that has the time and decision making authority to follow through each phase.

1. USER REQUIREMENTS DOCUMENT

Create a user requirements document that contains a quick list of one or two-line sentences identifying the functionality that is required for your business. This is not to be confused with a Request for Proposal (RFP). Typically, RFP documents are all inclusive and geared toward larger companies. Gather your validation team together and brainstorm what functionality you require. Examples of user requirements include: “System must be able to track lot numbers for raw materials and manufactured product” or “An item needs to be able to be tracked with different quality control status.” Preparing the user requirements document prior to your software selection process will help to ensure that you select a product that will best fit your needs.

2. PROJECT PLAN

In creation of the project plan you will identify who, what, where and when the validation will occur. A validation team will oversee the entire process, and typically includes the project manager, technical lead, quality assurance lead, validation lead and a support lead. Responsibilities of each team position should be explained in the project plan. The project plan should also include a system description, purpose, environment specifications, assumptions, exclusions, limitations, testing, acceptance criteria, error resolution and system documentation. There are various websites with samples and templates available to get you started.

3. FUNCTIONAL SPECIFICATIONS DOCUMENT

Create a functional specifications document that is an extension of the user requirements, but contains slightly more information. Each requirement now expands to three or four sentences. Let’s refer to the user requirements example of “System must be able to track lot numbers for raw materials and manufactured product.” The system will assign the internal lot number automatically upon receipt of a raw material. Manufactured item lot numbers will be assigned automatically based on the batch ticket number. A link is desired between the user and functional requirements as it will certify the testing prerequisites. The end result is an accountability matrix that will allow you to trace your tests back to the requirements.

4. GAP ANALYSIS

A review of the functional specifications occurs and a gap analysis is created. The gap analysis is used to determine risk, or the difference between the desired performance and the existing performance. These risks should be documented and prioritized. If a gap is identified as a high risk, it becomes important to ensure that your procedures will mitigate the risk.

5. INSTALLATION PROTOCOL

The installation protocol document outlines how the software should be installed. This would customarily be provided by your vendor. The document may also include hardware specifications and installation instructions. Typically, it will include areas to document verification of specifications for the hardware and software.

6. INSTALLATION REPORT

This document serves as evidence that the software was installed correctly according to the vendor recommendations and design. If your vendor performed the installation, they will provide you with this information.

7. TESTING PROTOCOL

The testing protocol document outlines the specific objectives, procedures, data sets, test scenarios, expected results and acceptance criteria for the system testing process. These protocols should test the software components your company will utilize. It is not necessary to test every setting available. The software vendor should have already tested the various setting combinations. The testing protocol should simply include evidence that a test was performed. Evidence might include screen shots and report print outs documenting the projected result. This information will be used to generate the testing report.

8. TESTING REPORT

The testing report is an outline of the testing that has occurred. Typically, it will include an executive summary of the test execution that addresses adherence to test procedures, acceptability of results, as well as any unexpected results.

9. SYSTEM RELEASE/GO-LIVE

The system release allows the software to be used in production. By this time, the users have been trained, data has been entered and business scenarios have been completed. The users can now begin using the software.

10. VALIDATION COMPLETE

Once the validation is complete, the system must be maintained in a validated state. Maintaining this state requires that standard operating procedures are in place for addressing problematic concerns and resolution, change control, record retirement, etc. If changes are required within the system, the changes should be reviewed and assessed for risk. Necessary changes should be authorized, documented, tested and approved before implementation occurs.

Conclusion

A key component in a validated system is documented evidence that the validation plan has been accomplished. When approached in an organized manner, software validation does not have to be an overwhelming task. By preparing a plan and following through its steps from beginning to end, software validation is manageable. For more information about creating a software validation plan, please contact ProcessPro at info@processproerp.com.

ProcessPro
A Division of Open Systems

1.800.457.3548
www.ProcessProERP.com