

# VAISALA

Using the  
ISPE's GAMP  
Methodology  
to Validate  
Environmental  
Monitoring  
System  
Software



# Introduction

Continuous Monitoring Systems (CMS) are used in the pharmaceutical industry to detect out-of-specification (OOS) conditions in manufacturing, processing and distribution environments. These modern, Web-based monitoring applications can also send email alarms to notify personnel to take corrective action before OOS conditions, such as extreme temperature or humidity, can have a negative effect on product quality and safety. Because a monitoring system can be considered an “automated system” we can manage this system using the Good Automated Manufacturing Practice (GAMP) guidelines published by the International Society for Pharmaceutical Engineering (ISPE). Specifically, let’s consider the ISPE’s publications: “The GAMP Guide for Validation of Automated Systems in Pharmaceutical Manufacture” and “GAMP 5: A Risk-Based Approach to Compliant GxP Computerized Systems.”

Maintaining environmental conditions within product

specifications is a critical part of GxP operations. Commonly, this involves an automated system providing continuous monitoring and real-time alarming. The conditions that drug products are exposed to must be accurately recorded to prove that the product was created, processed and stored within the correct parameters.

A CMS, like all software-based systems, has a life cycle. It starts at acquisition and installation, proceeds through release and maintenance, to the system’s eventual retirement. These roughly describe the Software Development Life Cycle (SDLC) which is the typical way to manage a GMP software. In this article, we will focus on the qualification and validation phases of the Life Cycle of a monitoring software. These phases are important because a CMS software can easily be forgotten; it generally runs in the background of a facility’s daily operations. However, monitoring system software should not be overlooked when it comes to validation. An inadequately qualified CMS can

result in unwanted observations at inspection time, and uncomfortable questions during customer audits. To ensure a fully GMP compliant software qualification, we recommend using the GAMP methodology as a reasonable and systematic guide to ensure your monitoring system software performs as expected throughout its life cycle.

Here we outline a ten-step guideline for applying the GAMP methodology to the validation of continuous monitoring system software. The goal of this article is to simplify the GAMP approach and highlight the particular steps that you can take to easily integrate your validation efforts into your existing quality management systems. We also strive to show how the effort level required in validation processes is heavily weighted upon monitoring system complexity (i.e. according to the GAMP System Categories). Overall, a GAMP approach to validation as outlined in this article should increase the lifespan, usability, and compliance of your CMS software.

## Key Terms

- **A User Requirements Specifications (URS)** document describes what the end user needs a system to do. The document can prioritize the requirements as mandatory, desirable, optional, or possible in future versions. Example: The system must prevent false alarms due to normal activities such as door opening.
- **A Functional Specification (FS)** document describes the functions of a system and how these functions satisfy the requirements in the URS. It also contains the methods for verifying that these requirements have been met. It does not define the inner workings of the system; rather, the FS describes interactions between the system and its end users.
- **A Traceability Matrix (TM)** is used to outline project requirements and ensure they are met. Traceability matrices are usually in the form of a table that is used to track requirements and/or specifications that must be tested. The matrix guides the development of testing documents, and should be verified after tests are completed to ensure that all system requirements have been adequately tested.

# Using GAMP to Validate Continuous Monitoring System Software

This is a **ten-step** process, with different pathways for different categories of systems (I.E.: classified according to GAMP 4 and/or 5), and each involves different levels of effort.



## Develop a User Requirements Specification (URS) Document

The first step in selecting an adequate CMS is to determine your needs by developing a User Requirements Specification document. Creation of this document should, ideally, happen before the selection of the CMS, although that is (unfortunately) not often the case. Creation of a URS document is the single most important element of the GAMP process. Repeat: Creation of a URS document is the single most important element of the GAMP process. Ideally, the URS is created BEFORE the system is selected because it is an important tool that we will use to determine if a candidate system is appropriate. It is the document that will describe the required functions of the system. The URS document can also identify the needs of multiple stakeholders to create a consensus in system selection.

The goal of the URS is to list the system requirements necessary to allow your CMS to align with and be included in your existing Quality Management System (QMS). Any gaps between the CMS and QMS increase the risk of non-

compliance. Fewer gaps between your monitoring system and your QMS equate less risk, in both compliance and product safety. A properly developed URS ensures that your new system will fit in with your existing quality processes.

Additionally, the process of creating a URS with multiple stakeholders can initiate discussions of entirely new functions and new, more efficient approaches to monitoring. This is to be expected. Creating the URS is an opportunity to be flexible, creative, and strategic in ensuring that the system you select will match the needs of your environments, your products, and QMS.

A typical URS for a monitoring system will include sections specific to the functions of a CMS, including: Sensors, Network, Utilities, Infrastructure, Security, Alarming, IT and other requirements specific to your facility or your product. The requirements included should be “SMART” – Specific, Measurable, Attainable, Relevant, and Testable. This last element should inform how you choose system requirements; if you create a system requirement that is not testable, it’s going to cause problems later on. Here are some examples of requirements, (note the use of the word “must”):

*When you have two monitoring systems working in parallel—a main monitoring system and a redundant set of sensors— how do you defend (to a regulator) that one system provides the “official” record of conditions, and the other system is only to provide redundancy of recording in case of failure in the main monitoring system?*

Some firms implement a BMS and CMS in parallel. Often this can signify to inspectors that your firm has a real commitment to continuity of records. Generally one system is declared the “system of record” and differentiated from the “control system.” However, the outputs from two different systems are quite different; often a BMS includes many kinds of sensors and controls that require custom programming. This customized programming makes the validation process, necessary for GMP, quite costly. A more cost-effective option can be an off-the-shelf CMS designed for GxP applications. This second system can provide the requisite documents for inspection and audit processes and be the “system of record.” In addition, many monitoring systems can include redundant recording, so that even in the event of power or network downtime, the records are continuous.





**The alarming system must...**

- have the capability to notify facility personnel when sensor readings exceed threshold values.
- have configurable delays from 0 to 60 minutes before alarm generation and notification.
- allow multiple high and low thresholds.
- communicate alarm states by SMS text, email, and phone.



Each of the requirements above are specific and testable. In practice the URS will be developed by a committee of stakeholders, each of whom will bring an area of expertise to the discussion. A benefit to involving stakeholders at this early, crucial step is that approval by stakeholders is generally easier if they've been involved in the process of defining the system requirements. There will be revisions, and likely more requirements than any one system can properly meet. This is to be expected. It can be helpful to document any requirements that are left unsatisfied for traceability. This will ensure transparency of process for any work-around solutions that must be created

to meet unfulfilled requirements. If you delete unsatisfied requirements, workarounds may not be properly documented and included in your QMS.

While system selection based on the needs of multiple stakeholders is necessarily a compromise, creating a URS that is based on a broad range of needs prior to shopping for a system increases the likelihood of finding the best match for your facility or application. Ideally, companies will drive innovation and creativity from system suppliers by developing their requirements based on the actual needs of their GxP applications, rather than based on what is available in the market.

**STEP 2**

**Begin Building a Traceability Matrix**

This is the tool that will organize the entire qualification effort, starting with system selection. The Traceability Matrix will track the requirements listed in the URS to ensure each requirement is represented by a corresponding function in the system. The matrix also helps to verify that each function is tested. Effectively a giant spreadsheet, you will use the first column for the requirements listed in the URS document, and fill in the remaining columns—Functional Specification, Configuration Specification, and Test Protocol— as you select and qualify your system.

REQUIREMENT	FUNCTIONAL SPECIFICATION	CONFIGURATION SPECIFICATION	TEST PROTOCOL
The system must prevent false alarms due to normal activities such as door opening.			

*What does it mean when a system supplier says something is “configurable”?*

Beware! Sometimes this is not the case if you are using some kind of graphical coding language that is provided within the system. Remember: the system supplier does not determine the GAMP software classification of their system. Just because they call their software “configurable” that doesn’t mean that some of your requirements won’t require some custom coding, which according to the ISPE, makes it a GAMP Category 5 system.

## FAQ



### Audit Vendors and Select a Product

The next step is to find a system that meets the requirements outlined in your URS. You will need to evaluate each potential monitoring system using your URS as a tool to determine appropriate fit with respect to your QMS. You may have multiple constraints to be considered along with your URS, such as your acquisition budget, the long-term cost of ownership, or the validation capabilities of your firm. For example, can you perform the system installation and operation qualification in-house, or will you need to commission that work from a contractor or the system vendor?

Your goal is to identify a shortlist of candidate systems for further examination. Once you have your shortlist, you will audit the vendors in two ways. You can audit their quality system and facility to evaluate their commitment to quality, and you can audit their CMS itself. With the second option, you will use your traceability matrix as a tool. Make a copy of the matrix for each system you audit and then compare the system capabilities against your own system requirements. The greatest differentiator of systems will be the software type, as defined by GAMP guidance.



### Determine Your Software Type

The ISPE has determined categories to classify software types; they created five categories to make them easy to identify. The key categories in regards to monitoring systems are:

- Category 3: Off-the-shelf
- Category 4: Configured
- Category 5: Custom

Note that the nomenclature changed slightly between GAMP 4 and GAMP 5. For the type of software we are going to refer to as “Off-the-Shelf” software, GAMP 4 called it “Standard” and GAMP 5 renamed it “Non-configured.” Both are Category 3 software types; often called “plug-and-play,” this type of software is designed to be used out of the box. It is easy to deploy, but should not require configuring beyond run-time configurations.

Run-time configuration refers to the simple set-up tasks that enable the system to operate, but do not change the business process. An example would be items that allow for entering a department and company name to report headers, and setting up default printers or user types.

The next type of software is Category 4, which in GAMP 4 is called “configured software” and in GAMP 5, “configured products.” These are systems that cannot be deployed out of the box because certain parameters need to be set to match your business processes before use. Examples include user-defined input strings for drop-down menus, and creating specific reports. Although we are doing configurations beyond run-time, there is no custom code. This means that the code in the software is not new: it is standard and has been thoroughly tested by the system supplier, thereby increasing user confidence.

Category 5 software is “custom software” in GAMP 4 and “custom products” under GAMP 5. This type of system generally refers to directly programmed systems that require coding. However, it also includes any systems that require any new code, even if that code was created using non-custom functions within the application. The custom code is bespoke to create new processes. Because the process is new, it has not been tested by the system supplier, and must therefore be thoroughly tested by the user. Examples range from truly bespoke one-of-a-kind systems, to Macros created in VBA in a Microsoft Excel application.

The ISPE went to great measures to create these categories because the differences in effort and cost are quite large, making this distinctive categorization a valuable tool for evaluating systems in terms of the resources they will require for validation, and for understanding how a new system will be integrated into a firm’s quality processes.

**STEP**  
**5**

## Develop a Functional Specification (FS) Document

Once you have your shortlist of candidate systems, you will create a Functional Specification (FS) document. This describes all of the functions of the software and how it will fulfill the requirements set out in the URS. The functional specification document for an “off-the-shelf” and “configured” system should be as specific and detailed as possible. A draft version is often available from the system vendor. The FS for a customized system may be vague, as the system does not yet exist. If you are the developer of a customized system, this is likely something you will need to provide.

As the functional specification documents are created or assessed, they may reveal new applications for the CMS system that can be added to the URS document.

Each requirement must be addressed by a function; each function is included in the trace matrix:

REQUIREMENT	FUNCTIONAL SPECIFICATION	CONFIGURATION SPECIFICATION	TEST PROTOCOL
The system must prevent false alarms due to normal activities such as door opening.	The system will have a configurable alarm delay function to prevent false alarms.		

The URS and FS documents won’t always match up precisely, and updating the trace matrix will confirm what requirements have (and haven’t) been met. It’s important to remember that not all requirements have the same level of importance; some will be essential and others simply “nice to have.” You may integrate this rating into a process of weighting the requirements with your stakeholders in order to prioritize the requirements by importance. If necessary, you may revise your URS with a statement regarding the items that are not functionally satisfied by the system. Remember to note the process or workaround that will satisfy the requirement.

It is now time to finalize your system selection. Just remember that the type of system you end up choosing—Category 3, 4, or 5—will affect how much overall validation is required. If you select a Category 3 system, no more specifications are needed and development of test documents can begin (Step 7). In the case of Category 4 or 5 systems, there are more documents needed, so on to Step 6. The majority of monitoring systems sold are Category 4. Category 5 monitoring systems typically contain devices and controllers from multiple suppliers, and custom code is required to allow the parts to communicate and be integrated into a fully functional system (BAS or BMS).



*When selecting a system, remember that resources required for validation will be commensurate with the system type.*

**STEP  
6**

**Develop Detailed Specification (DS) Documents**

Detailed Specification (DS) documents, describe how the proposed system needs to be configured or programmed to perform the functions identified in the FS. These specification documents aren't needed for Category 3 systems, as these are already in their final form.

For a Category 4 Configured system, the Detailed Specification document is known as a Configuration Specification (CS). The CS describes how the system will be configured to match its functions to the business process.

The actual configuration process usually occurs on-site after system installation, and may be performed by the system vendor.

For a Category 5 Custom system, the Detailed Specification document is known as a Detailed Design Specification (DDS). The system does not yet exist and still needs to be created at this stage. The DDS will describe exactly how the system functions, vaguely described in the FS, and how it will be structured and programmed. This can serve as an example of why the Category 5 systems require the most testing and documentation of all categories. Further discussion of the DDS is a specialized topic and outside the scope of this article.

The elements of the CS should now be recorded in the trace matrix beside the corresponding requirements and functions each configuration item is meant to satisfy. Note in the example below, the configuration specification is specific and describes in detail how the function will be configured, and what you must do to test the function.



*How is GAMP enforced?*

GAMP is a guidance... which means it contains suggested solutions from industry experts. It's a set of principles meant to outline methods that ensure pharmaceutical products are manufactured with the highest quality standards. One of the core principles of GAMP is that quality must be built into each stage of the manufacturing process.

Since GAMP has been used so much, it has become a best practice document... but it's not a requirement. Having said that, if you fail to implement recommendations of GAMP, you may expect to be questioned by an auditor to determine what you did instead and why. If you depart from industry accepted best practice as described by GAMP, be prepared to justify the departure.

**FAQ**

REQUIREMENT	FUNCTIONAL SPECIFICATION	CONFIGURATION SPECIFICATION	TEST PROTOCOL
The system must prevent false alarms due to normal activities such as door opening.	The system will have a configurable alarm delay function to prevent false alarms.	The alarm delay function will be configured for a 10 minute delay prior to alarm activation.	

**STEP**  
**7**

## Develop Testing Documents

Now that the system has been chosen and specific configuration determined (if necessary), development of the testing documents can begin. This is a necessary step for all categories of systems, and it is essential that the process includes every GMP item identified in the URS, FS, and CS documents. You can use risk assessment techniques to simplify this process. If it's not a GMP function within the software, there may be no reason to test it. This is where your S.M.A.R.T. requirements come into play, because that will help identify what is truly GMP-related.

The testing protocols should be entered into the traceability matrix to ensure that there is a test for every requirement. In our example matrix, Alarm Delay Testing has been added as our test protocol to ensure the 10-minute delay is correctly configured and functions as specified.

REQUIREMENT	FUNCTIONAL SPECIFICATION	CONFIGURATION SPECIFICATION	TEST PROTOCOL
The system must prevent false alarms due to normal activities such as door opening.	The system will have a configurable alarm delay function to prevent false alarms.	The alarm delay function will be configured for a 10 minute delay prior to alarm activation.	Alarm Delay Testing



The testing documents are similar for Category 3 and 4 systems, with really only a Performance Qualification (PQ) to distinguish them. For a Category 3 system, a PQ of software functions should not be required because all functions would have been fully tested in the Operational Qualification testing. Remember that the business processes cannot be changed for a Category 3 system, leaving no software functions to challenge in a PQ. So, for a Category 3 system, software validation requires only IQ and OQ documents, and for a Category 4 system, software validation will include IQ, OQ, and PQ documents. In comparison, testing will be quite extensive for Category 5 systems, including: code review, module testing, FAT, commissioning, SAT, IQ, OQ, and PQ documents. Note that every system type will need commissioning and SAT, as a normal part of the hardware installation. The takeaway message here is that the extent of work involved to test the different types of systems should heavily influence your choice of system—choose according to your needs balanced against your capabilities (especially in terms of validation).

### Icon Legend

Icon	System Type	Summary
	Off-the-Shelf	Typically easiest to validate, with limited functions and few changes possible.
	Configured	Tradeoff of moderately increased validation effort to gain increased functionality and ability to specialize.
	Custom	Massive increase in validation effort for a tailor-made solution.
	All	Size of hourglass = Validation Intensity.



## STEP 8

### Finalize the Traceability Matrix

The Traceability Matrix should have been updated at every step, based on the URS, FS, CS, DS and test documents. As you review your TM, you may notice tests that have no requirement; re-evaluate whether you need the test. Likewise, there may be requirements that can't be tested. Annotate this in your matrix; why can't this be tested? What will the workaround be?

#### Now it's time to do a final check:

- ✓ **URS** - Finalized and approved. All the URS requirements are included in the Traceability Matrix.
- ✓ **FS** - Finalized and approved. All the FS functions are included in the Traceability Matrix. Ensure that every requirement is addressed by a function.
- ✓ **CS** - Finalized and all configuration items entered into the Traceability Matrix. Ensure that a configuration is specified for every configurable function.
- ✓ **Test Protocols** - All tests written and approved. Ensure that every requirement is tested.
- ✓ **Traceability Matrix** - Complete, finalized and approved.
- ✓ **Now Test!**

## STEP 9

### Run System Tests

This is where the fun starts! All requirements need to be tested using the Traceability Matrix as a checklist. This is why it is essential to complete the matrix at every step. The systems will now be running in a real-life setting, so there are likely to be a few issues, hopefully only minor ones. Most of these will be resolved but if things really don't work, try revising the requirements, developing a workaround, or contacting the vendor to see if there is a fix. There may be a bug in the system; this will require a patch from the vendor.

## STEP 10

### Maintain the System under Change Control

Once the system is running a smoothly, validated, and released for use, it still needs to be maintained. This will ensure optimal function, compliance, and reduced risk, as well as a long system lifespan. Remember, the GAMP approach is a life cycle approach, which means maintaining the system until retirement.

The key maintenance steps for any automated system are:

- SOPs
- Training
- Calibration
- Validation
- Change control (ensuring that any changes are introduced in a controlled fashion)

These items are beyond the scope of this paper. However, you can find webinars on this topic here:

- [How to Validate your Monitoring System Software](#)
- [Maintaining Compliance for GxP Systems](#)

*"The webinar was excellent and many thanks! I wish more companies would help educate their customers like Vaisala. The knowledge gained from these webinars can be used in real-world application and put to work immediately..."*

*Pat,  
Calibration Specialist*

## Conclusion

Since 1991 the Good Automated Manufacturing Practice forum has been working to clarify and disseminate best practices in the correct use of computerized systems for regulated industries. Their internationally recognized guidelines have become trusted methodologies for validation and qualification of systems that affect the quality of drugs, biologicals and devices. We hope that the steps and categories outlined here present a simplified but applicable interpretation of GAMP's risk-based approach to software validation. The goal was to provide you with an illustrative guideline for properly validating and integrating monitoring system software into your existing quality management systems. For more information on Vaisala's Continuous Monitoring System, please visit [www.vaisala.com/lifescience](http://www.vaisala.com/lifescience).

# About the Author

**Paul Daniel, Senior Regulatory Compliance Expert at Vaisala**, has worked in the pharmaceutical, biotechnology and medical device industries since 1996. He has worked on a wide range of qualification projects, including: process, cleaning, shipping, laboratory equipment, packaging, software, network, and computer validation. He has extensive experience in applying the principles contained in FDA 21 CFR Parts 11, 210, 211, and 820 and has authored and executed validation protocols for pharmaceutical manufacturing and software validation. Daniel has a bachelor's degree in Biology (with honors) from the University of California in Berkeley.



# About Vaisala



Vaisala provides environmental monitoring, measurement and validation systems designed for the life science industries. Our solutions are built on expertise in the standards and regulations of pharmaceutical, biotech and medical device applications, including: cleanrooms, laboratories, and distribution centers. Headquartered in Finland (campus shown below), Vaisala has offices in Australia, Brazil, Canada, China, France, Germany, India, Japan, Malaysia, South Korea, Sweden, Great Britain, the United States and the United Arab Emirates.

Contact us at [sales@vaisala.com](mailto:sales@vaisala.com) or visit [www.vaisala.com/lifescience](http://www.vaisala.com/lifescience).

**VAISALA**

[www.vaisala.com](http://www.vaisala.com)

Please contact us at  
[www.vaisala.com/contactus](http://www.vaisala.com/contactus)



Scan the code for  
more information

Ref: B211370EN-B ©Vaisala 2019

This material is subject to copyright protection, with all copyrights retained by Vaisala and its individual partners. All rights reserved. Any logos and/or product names are trademarks of Vaisala or its individual partners. The reproduction, transfer, distribution or storage of information contained in this brochure in any form without the prior written consent of Vaisala is strictly prohibited. All specifications – technical included – are subject to change without notice.