Light for a Potentially Cloudy Situation: Approach to Validating Cloud Computing Tools

Michelle Miller and Nicola Zaccheddu

Software tools continue to evolve, offering improved user experiences through intuitive design, streamlined processes, and efficient execution. In addition to the tools themselves improving, the volume of those offered on the cloud as software as a service (SaaS) and platform as a service (PaaS) also is growing.[1]

These software distribution models provide numerous advantages for companies, including[2]:

- Removing the need to install and run applications on the company's data centers.
- Eliminating the associated cost and maintenance of installing and running applications on the company's data centers.
- Providing the benefit of automatic updates.
- Eliminating the burden of keeping software up to date among internal staff.
- Broadening accessibility through access of the SaaS applications via any Internet-enabled device.

Although these advantages decrease the burden on information technology (IT) staff and can streamline business operations, for medical device companies and other regulated industries, establishing and maintaining the validated state of these software tools is not a trivial endeavor.

The objectives of this article are to explore the topic of establishing and maintaining the validated state of cloud computing tools and to provide practical, implementable considerations and suggestions for doing so. The intent of this work is not to provide a comprehensive, one-size-fits-all solution that addresses all possible risks associated with cloud computing tools; instead, it is to present a high-level framework for validating them. Elements of this framework can be incorporated into a company's quality management system (QMS) to suit its needs. Of note, specific assessments around safety, security, and privacy of cloud computing tools should be considered when pursuing the use of cloud computing tools. However, details related to the strategy for these assessments are not covered in this article.

Challenge of Validating Cloud Computing Tools

In 21 CFR Part 820 from the Food and Drug Administration (FDA),[3] the regulations are clear in stating the requirement for the validation of software used in the automation of production or the quality system (whether for design, manufacture, distribution, or traceability, as further explained in the *Quality System Compendium*"[4]):

- 21 CFR 820.70(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."

Companies have been validating on-premises software for these uses successfully for decades. When the software is obtained and installed locally and change management is implemented, ensuring that the software is validated for its intended use and any changes are appropriately assessed and validated is straightforward. However, many of the touted advantages for cloud computing are the same characteristics that pose challenges when considering these features through the lens of this regulation. Because the software resides on the cloud and/or the hardware resides offsite, updates can be made unbeknownst to the user. Although most cloud computing companies execute some level of testing before deploying

***Michelle Miller**, BS, MS, is the director of global validation at Illumina, Inc., in San Diego, CA. Email: mmiller@illumina.com*

***Nicola Zaccheddu**, BS, MS, is a senior quality manager at Philips, in Eindhoven, the Netherlands. Email: nicola.zaccheddu@philips.com*
Corresponding author

software or hardware changes, it cannot be assumed that they will test everything that is of specific interest to each of their customers. In some cases, obtaining a detailed list of what has changed can be challenging for customers. In addition, since the FDA regulation requires that the validation activities and results shall be documented, obtaining detailed evidence of the software vendor's testing also may be difficult.

With the rise in the number of software systems and tools on the cloud that can be used as part of companies' QMS (e.g., defect management tools, complaint handling software, backlog management tools, learning management systems, low-code platforms, software development testing and validation tools), the discussion of how to approach the use of these is now more relevant than ever. Of course, the most conservative approach is to avoid the use of tools that are updated without the company's consent. Although this is a straightforward approach, whereby companies follow the same strategy they have historically, they are consequently "kicking the can down the road." Cloud computing platforms are becoming increasingly prevalent due to business drivers affecting both the software vendor and the customer. The ease of use, expediency of deployment and updates, and decrease in costs and overhead will drive increasing prevalence of these tools in the marketplace. In addition, with the proliferation of these types of tools, the availability of on-premises software for many new and innovative tools is expected to decrease over time. It is prudent to get ahead of this curve to ensure that companies have a strategy for implementing these tools in a compliant way, allowing them to stay current with state-of-the-art software offerings.

When considering the validation strategy for third-party cloud-based tools, one proposal is that companies revalidate the tool as part of each software update prior to release into their system. This option poses many challenges. For this to be done, a comprehensive list of changes would need to be obtained from the vendor with an environment for testing in advance of the changes being deployed. Although this may be an option provided by cloud computing providers specifically targeting customers in regulated industries, it is not a guarantee. However, even if these details can be managed, the timing and frequency of the changes also would need to be orchestrated such that all testing (in addition to security and privacy assessments) could be completed in advance of deployment. Some companies have seen success in coordinating quarterly software updates in this manner. However, for large software companies offering a multitude of cloud computing products to a large number of customers, this process would be untenable.

Identifying a Sustainable Approach

At this point, revisiting the meaning and intent of the Quality System Regulation, as found in the preamble to 21 CFR Part 820, would be helpful. In comment 136 in the preamble, the FDA states the following[3]:

- "The agency believes that it is necessary that software be validated to the extent possible to adequately ensure performance. Where source code and design specifications cannot be obtained, 'black box testing' must be performed to confirm that the software meets the user's needs and its intended uses. FDA emphasizes that manufacturers are responsible for the adequacy of the software used in their devices, and activities used to produce devices. When manufacturers purchase 'off-the-shelf' software, they must ensure that it will perform as intended in its chosen application."

In addition, forthcoming guidance from the FDA on computer system assurance will emphasize leveraging a risk-based approach and critical thinking in determining the testing strategy for software. With this in mind, a comprehensive strategy that builds assurance in the quality of the software and its ability to function reliably can be created.

First, whether pursuing SaaS/PaaS solutions is a worthwhile endeavor for your company should be determined. In making this determination, leveraging a risk-based approach to determine the potential impact of an unpredicted failure of the tool (e.g., resulting from a faulty update applied by the vendor), possible mitigation measures, and

residual risks is a helpful place to start. Several key points to consider in this process are discussed below.

Safety Class of the Device

What is the intended use of the cloud computing tool in the context of your medical device? What's the safety class of the device? In the event of a tool failure, how likely is it to affect the safety, security, privacy, and efficacy of your device? Of note, if the product is in the FDA category of a "multiple function device product," also consider whether (and how) the cloud computing tool affects "device functions" and "other functions."[5]

Vendor's Quality Processes

How much do you know about the vendor and the quality of its products and processes? This is critical information in the case of SaaS/PaaS tools because the vendor typically is able to apply direct changes to the tool without requiring any actions or approval from the customer. For this reason, doing due diligence on the vendor's quality and security processes builds awareness about its change control and testing practices. The best case scenario is that the vendor is open and willing to share its quality and security processes. Unfortunately, this rarely happens. If the supplier is not willing to share that information, leveraging publicly available information for your investigation can be helpful. Information such as other users' feedback (e.g., via forums and websites), history of previous changes/patches, frequency of updates, bug reports, security breaches, and data leaks can provide a good overview of how the vendor has behaved in the past and of the overall quality and security of its products.

Tool Update Schedule

How will tool changes and updates be deployed? The level of control a customer has on tool update timing, frequency, change notification timing, and testing opportunities depends on the type of tool and service offered by the provider and by the agreements/contract stipulated. In some cases, the vendor's policy is to keep all customers using the most current version of the tool; updates are directly applied to the tool and users cannot opt-out.

Ideally, the vendor offers the option of evaluating/testing the changes before they are applied to your system. In this scenario, even if all changes are required to be deployed, you have some time to determine how the change will affect your system, report bugs, and establish mitigation measures, if needed. When the vendor does not provide this option, establishing safeguard mechanisms is important to:

- Detect an update of the tool as soon as it's applied (e.g., by having automated scripts that regularly check changes of a tool's version). With this information, if unusual outputs or behavior are subsequently detected, pursuing an investigation focused on the tool update can be initiated more expediently.
- Execute regression testing (including security testing, when applicable) automatically as soon as a tool update is received, providing immediate detection to determine if the tool change has affected your software adversely.

After evaluating the considerations above, the risks and benefits of using the SaaS/PaaS tool in your system will be clearer. If suitable mitigations can be established for any risks identified, pursuing next steps for implementing the use of a cloud computing tool may be reasonable.

For the reasons outlined above, working closely with your procurement team to negotiate a service-level agreement and/or quality agreement with the vendor, if possible, is helpful. However, the strategy and process for this are outside the scope of this article.

Before initiating use of the cloud computing tool in production, you will need to perform an initial validation of the tool based on its intended use within your product or process. This can follow a standard validation process by which you document the requirements for the tool, evaluate possible failure modes and outline mitigations, and develop test cases accordingly. Of note, due to the fact that you don't have access to the software code, performing exploratory testing is beneficial and your validation testing likely will need to be black box

testing. Upon successful completion of testing, confirmation of expected results should be documented.

A few things should be considered when writing the validation summary report. Because this is a cloud computing tool, documentation of the specific version may not be possible. Therefore, making a note of this in anticipation of any future questions would be useful. Similarly, specifying that revalidation of each new version may not be possible due to dependency on the vendor's update policy and schedule is beneficial.

Framework for Change Management

After you have the assurance that the tool meets the intended use and performs reliably, consideration needs to given to how changes or updates to the tool will be handled. Failure to plan for these could result in serious consequences for your product, processes, and/or company, depending on the impact. Consider the scenario where the cloud supplier deploys an update that has an unexpected adverse impact on your medical software's production environment. In this situation, it would be best if you have a change control approach that allows you to determine what changes need to be applied to your system to respond to the tool update, then record, implement, test, and release those changes quickly. Having a framework that allows an expedient way to assess and address a variety of questions is recommended. Why is the tool update causing an issue? Are other products/versions also affected? What's the root cause? Is there an impact for the end users? Is there a safety, security, or privacy impact? What is the best path for resolution? Who should be informed (e.g., users/regulators)? What testing should be executed to ensure that the fix does not cause additional issues? Can a short-term solution be provided while permanent corrective actions are implemented?

Figure 1 and Table 1 provide a helpful framework and set of questions to leverage when managing changes to the cloud computing tool.

To streamline the change management process, the following practices are recommended.

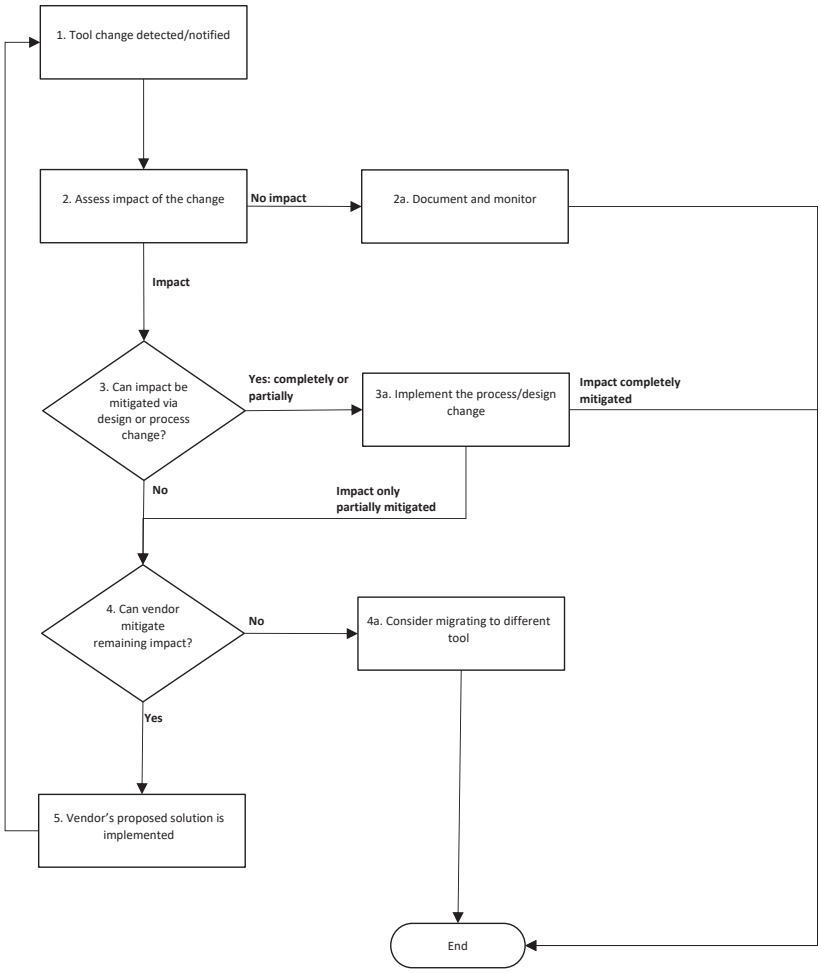

Figure 1. Change management framework tailored to cloud computing tools.

Documentation

Ensuring your documentation is clear and timely is important for maintaining internal records and traceability, as well as for facilitating questions that may arise in the future from external stakeholders. Automating the parts of your change management process that do not require human judgment will standardize the recording of important information, in addition to reducing burden on the team. One way to do this is to set mandatory fields in your change management tool.

Training

If change management activities are performed frequently, team members will be well practiced and skillful. However, if this is not the case, ensuring that staff are well trained and masters of the process will ensure that if and when an emergency change is needed, everyone is knowledgeable of the correct actions.

Continuous Improvement

Continuous improvement is applicable broadly across the business and relevant in change management. Through execution of continuous improvement and waste removal activities, the change management process can be kept lean and be optimized over time.

Even with great planning and a streamlined change management process, the possibility of an unexpected tool update that makes the system or product unusable should be considered. Creating a contingency plan for such a circumstance is advised, especially if the solution for the tool update requires substantial time to implement.

Thinking through options in advance and creating a workable solution streamlines the response should it be needed. Options could include creating a ready-to-implement transition plan that allows for prompt migration to another tool (e.g., on premises), so that your system can keep running until the issue is fixed. Alternatively, when feasible, redundancy could be strategically implemented into the system, so that an "emergency mode" is available that can be enabled, if needed. This mode would allow your system to perform basic critical functions in the absence of the cloud tool. Documenting the contingency plan in advance will allow the team to efficiently execute should it be needed.

Conclusion

Numerous business benefits are associated with leveraging cloud computing tools, making these tools appealing to implement. For many regulated companies, transitioning to these tools has been a cause for apprehension due to the perceived complexity of testing and implementing them in a compliant way. However, by leveraging the framework described here, the proper controls can be established to ensure compliance. This framework involves assessing the cloud computing tool to ensure that it meets the needs of the company, evaluating the risk to product and/or processes associated with the tool, conducting black box testing to establish confidence in the tool's performance, and setting up mechanisms to manage, test, and mitigate changes to the tool.

Step	Description
1. Tool change detected/notified	If the vendor of the tool does not provide advance notice of changes, putting in place mechanisms to increase awareness of changes is advised. These may include: (1) Proactively monitoring and reviewing information on user forums, including both official and unofficial channels. These forums often provide information and insights about bug/issues reported, data leaks, and planned changes. (2) Creating scripts to detect changes in the software as described in the TOOL UPDATE SCHEDULE section.
2. Assess impact of the change	After a change has been detected/notified, leverage the information available about the change and assess the impact of tool changes on the software system, process, or product (e.g., safety, security, privacy). Note: Having an up-to-date, readily available, and easy-to-navigate traceability matrix that indicates what cloud computing tool is used in each product/version, process, or system will help to quickly identify which systems should be assessed.
2a. Document and monitor	If there is no impact, documenting the change to the software and monitoring for any unexpected behavior or output from the tool is recommended.
3. Can impact be mitigated via design or process change?	If there is an impact, determining whether it can be mitigated through a design or process change on your side is recommended.
3a. Implement the process/design change	If the impact can be completely or partially mitigated via design/process change, move forward with your company's process for implementing the design or process change. This should include verifying or validating the change. This testing should also include regression testing, as described in the second bulleted item in the TOOL UPDATE SCHEDULE section.Following successful testing, the design or process change can be deployed/released. Communicating the design change to the necessary internal and external stakeholders (e.g., regulators, users, other businesses) is of vital importance.
4. Can vendor mitigate remaining impact?	If vendor action is needed to mitigate the impact of the tool change, having an open communication with the vendor is it's recommended. Explain what challenges are being encountered and solicit the vendor's input on next steps. If mitigations on the vendor's side will take longer than deemed appropriate for your business needs, a short-term alternate solution may be beneficial to implement while the vendor devises a long-term solution.
4a. Consider migrating to a different tool	If the software tool change cannot be mitigated on your end through a design or process change or through action by the vendor, your company should consider whether it would be better served by moving to a different tool.
5. Vendor's proposed solution is implemented	After the solution on the vendor's side has been implemented, go back to step 1 and evaluate the new change.

Table 1. Change management framework steps description.

References

1. Gartner. Gartner Forecasts worldwide public cloud revenue to grow 6.3% in 2020. www.gartner.com/en/newsroom/press-releases/2020-07-23-gartner-forecasts-worldwide-public-cloud-revenue-to-grow-6point3-percent-in-2020. Accessed April 5, 2021.

2. National Institute of Standards and Technology. *The NIST Definition of Cloud Computing*. SP 800-145. https://csrc.nist.gov/publications/detail/sp/800-145/final. Accessed April 5, 2021.

3. Food and Drug Administration. *Medical Devices; Current Good Manufacturing Practice (CGMP) Final Rule; Quality System Regulation*. www.fda.gov/medical-devices/quality-system-qs-regulationmedical-device-good-manufacturing-practices/medical-devices-current-good-manufacturing-practice-cgmp-final-rule-quality-system-regulation. Accessed April 5, 2021.

4. Association for the Advancement of Medical Instrumentation. *Quality System Compendium: GMP Requirements & Industry Practice*. 4th ed. Arlington, VA: Association for the Advancement of Medical Instrumentation; 2019.

5. Food and Drug Administration. *Multiple Function Device Products: Policy and Considerations*. www.fda.gov/media/112671/download. Accessed April 5, 2021.