EXECUTIVE SUMMARY

Information Systems in Clinical Trials: Pragmatic Validation of EDC Systems

A practical approach to the validation of electronic data capture (EDC) information systems

OVERVIEW

For computer systems used in relation to clinical studies, validation is a key regulatory requirement that challenges the understanding and resources of many organizations. Achieving the validated state means managing an ever-changing environment of servers, operating systems, electronic data capture (EDC) software, security patches, and more, some or all of which may be outside the direct control of the organization, and all of which require consideration before even starting to develop the clinical trial study database. To meet regulatory requirements for computer system validation (CSV), a pragmatic, risk-based approach may be taken. This article explores the scope of EDC system validation as well as offers practical ways of dealing with change overload and for surviving a regulatory inspection.

REGULATORY EXPECTATIONS FOR CSV

CSV is not just testing. Rather, CSV is performed to demonstrate that the whole system, including software, hardware, process, and people, performs consistently and is fit for purpose. CSV starts with developing a validation plan that describes the testing process—what will be tested, why the testing is needed, and how the testing will be performed. The validation plan will outline the testing required to demonstrate that the system maintains data integrity and, for systems that perform drug delivery, titration, allocation, or safety monitoring, to show that the system performs properly to ensure patient safety will not be compromised. In addition, the validation plan must describe how the system meets...
EXECUTIVE SUMMARY

While there is a clear regulatory expectation that computer systems must be validated, the process for doing this is highly subjective. Therefore, a risk-based approach to validation is recommended. This can be accomplished by following the principles of good automated manufacturing practice (GAMP), an extremely useful set of principles and procedures that was created to address evolving regulatory agency expectations for computerized system compliance and validation, and to facilitate the interpretation of regulatory requirements. The main premise of GAMP is that software may be risk classified, where the degree of validation performed may be proportionate to the degree of risk. Custom applications developed to meet specific needs would present the highest risk, whereas applications such as operating systems that are purchased off the shelf and used as is without changing any configuration would present the lowest risk.

GAMP outlines a prospective software development life cycle (SDLC) approach following a “V-model”, where execution of validation processes occurs in a sequential manner that can be represented by a V-shape (Figure 1). The V-model summarizes the main steps to be taken in conjunction with the corresponding deliverables. The left side of the model represents the project definition, starting with documentation of the user requirements, which are then translated into functional specifications, and then into design specifications. The system is then constructed based on these requirements. The right side of the model represents project testing and integration, starting with installation qualification (IQ) to demonstrate that the system meets design specifications, followed by operational qualification (OQ) to demonstrate that the system meets functional specifications, then
by performance qualification (PQ) to demonstrate that the system meets user requirements when operating in the actual working environment.

EDC SYSTEM VALIDATION PROCESS

Roles and responsibilities. A typical EDC system arrangement (FIGURE 2) consists of the clinical study database (i.e., Study Build), which is built in the EDC software application that is hosted as software as a service (SaaS) at a remote data center, possibly by a third-party service provider. While the initial setup of this arrangement is fairly straightforward, each of the components may need to be modified during the course of the clinical study. For example, the clinical investigator may need to add, modify, or remove case report forms (CRFs), and the EDC application may require updating as software patches become available. With respect to the hosting environment, the server(s) will require maintenance and firmware and/or software updates may need to be performed. Therefore, to ensure that the validation is completed with an appropriate degree of due diligence and that the system continues to operate in a validated state in a constantly changing environment, it is essential to clearly delineate and define responsibilities for each of the parties involved. Typically, a clinical trials unit (CTU) or clinical research organization (CRO) will be responsible for building and maintaining the study database. They will purchase the EDC application software as a service from an EDC vendor, who will likely have the application hosted by a separate hosting provider (e.g., Amazon Web Services or Rackspace Technology). Overall accountability will be the responsibility of the sponsor (FIGURE 3).

Procurement. The first part of the validation process is procurement. To ensure the EDC system will be appropriate for the intended use, the procurement process should include a product assessment, a vendor assessment, and a hosting assessment.
EXECUTIVE SUMMARY

The product assessment (FIGURE 4) starts with determining functional requirements. This can be a long and tedious process, but it is important to document everything the system needs to do, including saving and editing data, audit trail functionality, and querying capabilities. Accurate documentation of functional requirements is critical to properly assess the software during demos. Next is the determination of non-functional requirements, such as security, backup, and performance, which should be evaluated with respect to the number of users and the number of studies it is anticipated the system will be used for.

Then there are service requirements, which involves understanding the expected uptime, downtime (planned and unplanned), and how any related notifications are rolled out. Finally, it is useful to prepare a product roadmap to document when any expected patches will be made available, and it is helpful to obtain access to any relevant user groups that may be a source of independent verification of the application’s performance quality.

When performing the vendor assessment (FIGURE 5), find out how long the vendor has been in business and determine whether they are financially stable. Review their organogram to see whether they have adequate staffing. Also review published reference cases or obtain references from other users. Ask the vendor to describe their quality system to determine whether it is adequate, and evaluate their standard operating procedures (SOPs) to confirm that they have an appropriate SDLC process (e.g., determine whether they are familiar with and following GAMP). Review the documentation for the last few software releases to verify they are following SOPs and are capable of producing high-quality validation documentation. And finally, assess their processes for and frequency of bug and hotfix management.

The hosting assessment includes many of the same activities as the vendor assessment (FIGURE 6). Typically, the EDC vendor is responsible for contractual arrangements with the hosting vendor. Ask the EDC vendor for their vendor assessment of the hosting company. In addition, review the service-level agreement (SLA) between the EDC vendor and the hosting vendor to review the details of the services the host will provide with respect to your system, including data protection and backup (e.g., where the data and backup data are stored, who has access, and on what schedule the backup and restore of data will be tested).

Computer system qualification. The next step of the validation process is to perform the IQ, which is testing of the software installation, followed by OQ, which is testing of the functional requirements, and then by PQ, which is testing of both the functional requirements and the non-functional requirements.
EXECUTIVE SUMMARY

**Figure 7: Computer system qualification.**

- IQ - software installation
  - Vendor should do this, following standard process
  - Probably already done (shared tenancy)
- OQ - does it do what it should?
  - Systematic testing of all functionality, huge job, often (partially) automated
  - Vendor should do this, following standard process
  - Bug list often useful proxy for quality
- PQ - does it do what it should "for you"?
  - Real-life testing, your actual instance
  - Functional and non-functional

In the actual production instance (**FIGURE 7**). Assuming your company is not purchasing their own EDC system, the EDC vendor should perform IQ and OQ. As part of the vendor assessment process, be sure to review the EDC vendor’s SOPs to ensure they are following a standard process that includes implementation of a validation plan, then review the executed IQ documentation to verify that they followed their validation plan and SOPs. The IQ and OQ are typically executed in a test environment. The PQ should be executed in the production instance by staff members who will be using the system.

An important consideration of the PQ testing is whether it is necessary to repeat all of the testing performed during OQ, especially when performing revalidation when a new version of the software is released, or when software patches will be applied. Using a risk-proportionate approach, the degree of testing performed under PQ can be greatly reduced, as long as there is confidence in the vendor assessment process and in the IQ and OQ performed by the EDC vendor, and as long as the IQ and OQ documentation will be readily available during a regulatory inspection. Whenever a new version release or patch is released, review the release notes and assess the risk for each individual component, then design the testing based on what really needs to be checked based on the risk to the validated state of the system.

**REGULATORY INSPECTION CONSIDERATIONS**

To ensure that the system validation will meet the expectations of a regulatory inspector, it is important to consider these expectations when developing the validation plan. First, the most common issue that arises during an inspection is that companies frequently do not have a documented validation plan—they simply determine the system specifications and then test against those specifications. A formal, written validation plan is essential. Most importantly, the inspectors will review the validation plan to ensure it includes the justification used to determine the qualification testing. In addition, the expectation is that the validation plan will be reviewed and updated at least annually, even if no changes have been implemented. Finally, it is very useful to rehearse the inspection to practice how the validation plan and overall strategy will be presented to an inspector.

**CONCLUSION**

Validation of EDC systems is complex as a result of the ever-changing environment in which they operate. To meet regulatory requirements for CSV and to prepare for a regulatory inspection of the system validation, it is useful to follow a risk-based, pragmatic approach to validation. It is important to be familiar with the regulatory expectations for validation strategy and documentation, and it is essential that the strategy be outlined and justified in a formal, written validation plan.