COMPUTER SOFTWARE ASSURANCE and the Critical Thinking Approach

By Gaurav Walia and Danilo Neri, PhD

In 2022, the US Food and Drug Administration (FDA) issued their draft guidance "Computer Software Assurance for Production and Quality System Software" [1] to enhance the computer validation process required by predicate rules, either in the pharmaceutical or medical device space. The critical thinking approach was introduced by ISPE GAMP® Guides and emphasizes a focus on clear thinking through a plan, then creating documentation from a process perspective. These methods combined create the optimal replacement for computer system validation (CSV).

New FDA draft guidance marked a transition to computer software assurance (CSA) in 2022 [1]. Historically, CSV practices unfortunately focused on generating great amounts of paper-based testing records, and it was believed that the larger the stack of documentation, the better the quality and effort when validating a computerized system and eventually presenting it to the FDA.

The reality is that a mountain of paperwork did not equate to proper CSV, which resulted in a failure to fully ensure the product quality, data integrity, or patient safety. Critical thinking is a term that was introduced by GAMP guidance (initially in *ISPE GAMP® Good Practice Guide: Enabling Innovation* [2] and then more recently in *ISPE GAMP® 5: A Risk-Based Approach to Compliant GxP Computerized Systems (Second Edition)* [3]), and it focuses on those critical considerations missed by CSV.

When applied, this critical thinking approach potentially results in the conclusion that a tremendous amount of documentation cannot possibly represent quality, especially when inspectors continue to identify a multitude of issues during inspections. These issues stem from items including, but not limited to, segregation of duties, data integrity potential violations, and whether the system works as intended after testing.

The FDA draft guidance states [1]: "Computer software assurance is a risk-based approach for establishing and maintaining confidence that software is fit for its intended use. This approach considers the risk of compromised safety and/or quality of the device (should the software fail to perform as intended) to determine the level of assurance effort and activities appropriate to establish confidence in the software. Because the computer software assurance effort is risk-based, it follows a least-burdensome approach, where the burden of validation is no more than necessary to address the risk. Such an approach supports the efficient use of resources, in turn promoting product quality."

UTILIZATION OF THE CRITICAL THINKING APPROACH

The critical thinking approach can be used in potentially limiting testing evidence for low-risk functionalities through the following use of innovative testing approaches: unscripted and scripted testing.

Unscripted Testing

This is dynamic testing in which the tester's actions are not prescribed by written instructions in a test case. This can include:

- Ad hoc testing: A concept derived from unscripted practice that focuses primarily on performing testing that does not rely on large amounts of documentation (e.g., test procedures) to execute
- Error-guessing: A test design technique in which test cases are derived on the basis of the tester's knowledge of past failures or general knowledge of failure modes
- Exploratory testing: Experience-based testing in which the tester spontaneously designs and executes tests based on the tester's existing relevant knowledge, prior exploration of the test item (including results from previous tests), and heuristic logical rules regarding common software behaviors and types of failure

Scripted Testing

This is dynamic testing in which the tester's actions are prescribed by written instructions in a test case. It includes:

- Robust scripted testing: Scripted testing efforts in which the risk of the computer system or automation includes evidence of repeatability, traceability to requirements, and auditability
- Limited scripted testing: A hybrid approach of scripted and unscripted testing that is appropriately scaled according to the

risk of the computer system or automation. These test methods require that the tester, through training and/or experience, has the knowledge to devise appropriate tests

WHY SOFTWARE ASSURANCE IS CRITICAL

Software assurance saves time and money. Software quality assurance ensures that developers find bugs and errors at the early stages of software development when less amount of time and money is required to fix them. The CSA approach provides flexibility and agility in helping ensure that the software is maintained in a validated state.

In determining the risk-based approach for software systems, the main intent is to use a risk-based methodology to ensure targeted and focused validation. This confirms that the computerized system is functioning for its prescribed, intended use, focusing (i.e., limiting a larger amount of documented evidence) on those functionalities that may directly impact patient safety and product quality.

For example, a software feature, function, or operation might pose a high process risk when its failure to perform as intended may result in a quality problem that foreseeably could compromise patient safety. In addition, software features and functions may maintain process parameters—such as temperature, pressure, or humidity—that affect the physical properties of product or manufacturing processes which are identified as essential to product safety and quality.

THE CSA RISK FRAMEWORK

The CSA risk framework is a risk-based approach intended to help manufacturers establish and maintain the reliability and safety of computer software throughout the life cycle, reducing the testing effort. Several steps must be taken to ensure the successful implementation of the CSA approach.

Identifying the Intended Use

Manufacturers must first determine whether the software is intended for use as part of production or the quality system. In general, software is either used directly as part of production or the quality system, or it supports production or the quality system.

Software that is used directly as part of production or the quality system includes software intended for automating production processes, inspection, testing, or the collection and processing of production data. It also includes software intended for automating quality system processes, collection, and the processing of quality system data, or maintaining a quality record established under the quality system regulation.

Manufacturers must validate software with these intended uses to ensure that it is reliable and functions as intended according to the associated risk. This is critical for both the pharmaceutical and biopharmaceutical industries, as well as medical device manufacturers.

Determining the Risk-Based Approach

The CSA risk framework provides guidance on identifying and mitigating the risks associated with software validation and includes examples of applying the framework to various CSA situations. Once it has been determined that a software feature, function, or operation is intended for use as part of production or the quality system, a risk-based approach should be used to determine the appropriate assurance activities.

The CSA risk framework is a risk-based approach intended to help manufacturers establish and maintain the reliability and safety of computer software throughout the life cycle, reducing the testing effort. Several steps must be taken to ensure the successful implementation of the CSA approach.

This approach involves identifying historical software failures, determining whether such a failure poses a high process risk, and systematically selecting and performing assurance activities commensurate with the medical device or process risk. The risk-based analysis for production or quality system software should consider which failures are historically known (as opposed to likely) and the risks resulting from each such failure.

Process risks refer to the potential to compromise production or the quality system, whereas direct system risks refer to the potential for a device to harm the patient or user. In CSA, the risk determination is therefore a key element to be carefully designed and executed to focus validation testing on the functionalities oriented to manage high-risk processes.

Determining the Appropriate Assurance Activities

Different types of assurance activities should be considered, corresponding with the system risk or process risk. For software features, functions, or operations that pose a high process risk, assurance activities should be considered higher corresponding with the system risk. Conversely, for software features, functions, or operations that do not pose a high process risk, appropriate assurance activities should be considered lower corresponding to the risk identified.

Table 1: Comparison of CSV to CSA (part 1).

CURRENT VALIDATION	CSA
Focus on creating documents/records for compliance and securing evidence for auditors (paper mountain effect)	Focuses on critical thinking and testing to achieve higher confidence in the quality of the system
Risk is based on failure mode effects analysis (FMEA) business risk and meeting regulatory requirement	**Risk is based on the impact on patient safety, product quality (intended purpose), and data integrity**
Validation involves testing all functionality (e.g., focus on COTS vs. potentially missing the higher-risk areas)	Validation is based on applying the right amount of diligence to a given level of risk to patient safety, product quality, and data integrity
Ignores previous assurance activity or related risk controls (e.g., one example is COTS functionalities)	**Leveraging prior assurance activity, risk controls, and any documentation that exists with the vendor and others** (e.g., build a good vendor risk management/vendor audit program that can be used to justify reduced COTS function testing)
All testing is scripted	Uses unscripted and ad-hoc testing in addition to scripted testing and employs the least burdensome approach to documentation
80% of time spent on documentation / 20% of time spent on testing	20% of time spent on documentation / 80% of time spent on testing

CSA AND REQUIRED RECORDS

CSA enforces the need for regulated companies to establish the appropriate record. When establishing the record, the manufacturer should capture sufficient objective evidence to demonstrate that the software feature, function, or operation was assessed and performs as intended. In general, the record should include:

- The intended use of the software feature, function, or operation
- The determination of risk of the software feature, function, or operation
- Documentation of the assurance activities conducted
- The date of testing activities and the name of the person conducting them
- Establishing review and approval, when appropriate

The documentation of assurance activities should not include more evidence than necessary. However, it should retain sufficient details to serve as a baseline for improvements or as a reference point if issues occur. To reduce the need for manual or paper-based documentation, the FDA recommends that manufacturers "leverage automated traceability, testing, and the electronic capture of work performed to document the results. As a least burdensome method, FDA recommends the use of electronic records, such as system logs, audit trails, and other data generated by the software…in establishing the record associated with assurance activities" [1].

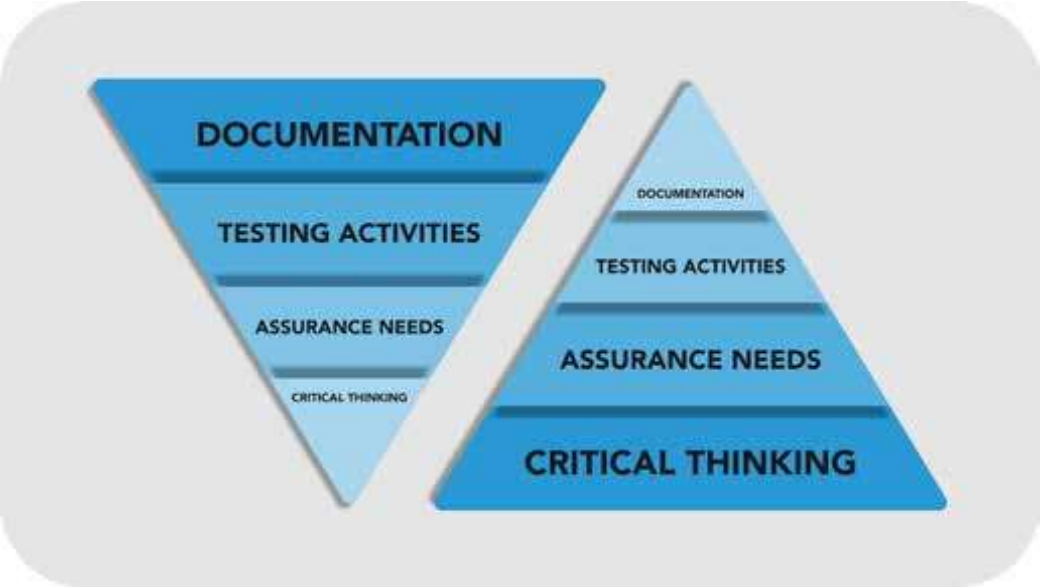
THE DIFFERENCE BETWEEN CSV AND CSA

CSA is a new way of looking at the traditional CSV approach through consideration of risk whereby the focus is on what matters—patient safety, product quality, and data integrity. A comparison of the two approaches for assurance activities and records is shown in Table 1.

The focus of CSV is a decision to carry out excessive and unnecessary tasks/testing, which creates excessive documentation, and too much time is spent on testing on all system functionalities. In comparison, when using CSA, the primary focus allows manufacturers to leverage principles such as risk-based testing, unscripted testing, continuous performance monitoring, and data monitoring, as well as validation activities performed by other entities (e.g., developers, suppliers). The CSA approach provides flexibility and agility in helping assure that the software maintains a validated state. This allows for assurance that the data provided is accurate, compliant, and more efficient.

In Figure 1, the left triangle represents the CSV approach, and the right triangle shows the differences that occur with CSA. As shown, in CSV, the decision to carry out excessive and unnecessary tasks/testing with documentation is the primary function, whereas critical thinking comes last. Issues and potential enhancements are identified in the last stages of the processes, resulting in a need for reworking and a waste of resources.

Figure 1: Comparison of CSV to CSA (part 2).



Current thinking with respect to *GAMP® 5 Second Edition* promotes a risk-based approach to ensure fitness of the system for its intended use [3]. In many instances of validation strategy, people do not apply critical thinking to ensure that the approach they are taking is customized and proportionate to the needs of the specific system they are validating: for example, when implementing the 21 CFR Part 11 requirement of segregation of duties (SOD), where roles and privileges are properly defined, configured, and followed [4, 5].

Using critical thinking (instead of having general training on the system regarding SOD), a customized/tailored approach to system training around the routine tasks of the different system user types should be followed. This will ensure that each user type is given training centered on their specific tasks and responsibilities so that SOD is clearly defined and followed.

Critical thinking is the most advantageous process for companies and manufacturers. It ensures that issues and enhancements are determined in the first stages, thereby focusing proper efforts toward assurance needs. Thus, it results in better-defined testing activities to ultimately ensure proper intended use and working functionality of the computerized system.

TRANSITIONING FROM CSV TO CSA

Organizations that want to pivot from a CSV to a CSA approach need to properly invest in activities and apply critical thinking to how software assurance will be performed and to how time and efforts will be allocated to achieve these tasks. For example, if a firm is implementing a new laboratory or manufacturing software/computerized system, the firm should apply critical thinking to assess the system and functional requirements risks as well as the intended use of the system, instead of spending significant time on excessive testing or documentation. This is because risk determination is the CSA core process.

The computerized system should be explored for what its intended use will be, what the key functions are, and what risks lie with its functionalities. Its potential impact on product quality and patient safety, historical issues, data integrity, and potential violations should also be assessed. This is necessary to properly outline the strategy of assurance and testing activities before the true efforts on documentation begin.

In addition, after the initial critical thinking phase of the process, teams should decide the key software assurance activities that must be performed to determine what needs to be tested and how testing should be documented. This will determine the amount of testing and evidence required to prove that the test function is working as intended.

The culmination of the activities previously mentioned should then be combined with a proper vendor risk management inspection program—especially for information technology service providers and technology vendors—that should include equipment and software-computerized systems.

Vendor risk management, audits, and inspection programs currently represent significant industry gaps. Performing these activities and having vendors' quality systems assessed, when in good standing and documented, can potentially be further leveraged. This is especially true with commercial-off-the-shelf (COTS) and software as a service (SaaS) platforms when leveraging these vendor audits/inspections to minimize out-of-the-box functionality testing or excessive service testing/verification.

The activities previously mentioned are a recommended way of pivoting from CSV to CSA in practice—and from a CSA perspective. It is equally important to shift and develop an organization's mindset and culture to change from CSV to a CSA approach with critical thinking.

Having a proper quality system with a good foundation that establishes a vendor risk management or vendor audit program is a necessity.

The risk-based analysis for production or quality system software considers factors that may impact or prevent the software from performing as intended, such as proper system configuration and management, security of the system, data storage, data transfer, or operation error. Thus, a risk-based analysis for production or quality system software should consider which failures are reasonably foreseeable (as opposed to likely) and the risks resulting from each such failure.

Pivoting from CSV to CSA and using critical thinking can help the industry become more compliant and drive processes to be more agile while also ensuring better data integrity, product quality, and patient safety. It will also concurrently ensure that a stronger focus on validation of computerized systems will come with a more focused risk-based approach. Industry examples of CSA implementation while demonstrating new strategies with effective critical thinking can help ensure that the new paradigm will result in also using a risk-based approach to work smarter and not harder.

KEY STRATEGIES FOR CSA IMPLEMENTATION

Key strategies for effective implementation of better computerized systems with more efficient and agile processes and less work can produce more targeted efficiency with improved outcomes for product quality, patient safety, and data integrity.

As an example, COTS spreadsheet software may be composed of various functions with different intended uses. Based on FDA guidance on CSA, when using the basic input functions of the COTS spreadsheet software for the intended use of documenting the time and temperature readings for a curing process, a manufacturer may not need to perform additional assurance activities beyond those conducted by the COTS software developer and initial installation and configuration [1].

The intended use of the software—documenting readings—only supports maintaining the quality system record and poses a low process risk. As such, initial activities such as the vendor assessment and software installation and configuration may be sufficient to establish that the software is fit for its intended use and is maintained in a validated state.

However, if a manufacturer uses built-in functions of the COTS spreadsheet to create custom formulas that are directly used in production or the quality system, then additional risks may be present. For example, if a custom formula automatically calculates time and temperature statistics to monitor the performance and suitability of the curing process, then additional validation by the manufacturer might be necessary. Therefore, having a proper quality system with a good foundation that establishes a vendor risk management or vendor audit program is a necessity.

Using the SaaS Model

The increased utilization of SaaS solutions with cloud computing combined with proper and actual vendor audits (i.e., focused to verify the current practices of the vendor and not limited to high-level verifications) and a third-party vendor risk management program produces benefits. It allows for leveraging within the CSA approach (with SaaS solutions that have a good audit history) and for a reduction of the overall scope of SaaS solution and COTS functionality validation testing.

Manufacturers are responsible for determining the appropriate assurance activities to ensure the software features, functions, or operations are in a validated state. Some possible assurance activities and considerations (such as leveraging vendor risk management programs or third-party vendor audits, e.g., in SaaS or cloud computing vendors) reduce the amount of testing while leveraging any of the aforementioned activities. This is potentially combined with historical vendor testing data, COTS functionality, and overall historical data/experience with the software or vendor.

Benefits of the SaaS model

The SaaS model of selling software has become increasingly popular over the years due to its many advantages. The SaaS model allows customers to pay for only the services they use, eliminating upfront costs associated with purchasing and maintaining physical infrastructure or exhaustively upgrading on-premise versions.

Advantages of cloud technologies

The benefits of cloud computing for pharmaceutical companies are numerous, and include faster time to market, scalability, flexibility, cost savings, better collaboration, advanced security, and data loss prevention. In addition, new instances can be implemented, or old instances can be retired in seconds, allowing developers to accelerate development with quicker deployments.

Cloud computing enables the pharmaceutical community to innovate rapidly, manage changes effortlessly, and deliver new medicines to the market faster. Cloud-based infrastructure offers secure storage for sizable and sensitive information. It also supports data security, integrity, and compliance with regulatory entities.

Key Metrics for Success

An important step to determine successful implementation and delivery of a CSA approach is to ensure proper formulation and monitoring of key metrics or key performance indicators (KPIs). Examples include:

- Return on investment: Cost savings for overall reduction in the duration needed to ensure the software is working as intended
- Writing: Significant reduction in the writing of test scripts and protocols
- Execution: Significant reduction in the execution and testing of test scripts and protocols
- Review of key deliverables: Significant reduction in the review of key deliverables
- Approval of key deliverables: Significant reduction in the approval of key deliverables

The overall benefits include a shorter duration for testing with quicker approval times and less production issues. This also potentially produces fewer issues during an audit or inspection where the system can be shown to be tested properly to ensure it is functioning correctly. Thus, it also potentially reduces risk in data integrity, product quality, and patient safety.

GAMP® 5 Second Edition provides further detailed guidance on how to apply CSA and critical thinking. The opportunities and concepts discussed in the FDA draft guidance on scope of testing and the use of the most appropriate types of testing can readily be adopted and implemented within the system life cycle approach described in *GAMP® 5 Second Edition*.

CONCLUSION

Critical thinking emphasizes a focus on, first, clearly thinking through a plan and then creating documentation from a process perspective. Although this critical thinking theory started with medical devices, it can apply to any pharmaceutical or biotechnology product. Instead of companies generating mountains of paperwork that "say" the product is safe, CSA can reduce the amount of testing and documentation. It offers greater efficiency, creates fewer steps, and ensures safer products, and, therefore, better patient outcomes.

Digital systems are increasingly being proliferated by industry, and manufacturers must eventually become up to date with technology. Electronic data can be more readily accessible, retrieved, analyzed, and presented with more efficiency, especially during an audit or inspection. This is another benefit of a CSA approach.

CSA, with GAMP in mind, starts with critical thinking instead of first developing the documentation; this can work for paper documents as well as computer system digital formats. The focus is on quality aspects—assurance means more focus on patient safety and data integrity, which is where the most risk lies. Documentation should then be created accordingly. Rather than focusing on paper-based exercises, which can create problems, the focus should be on thought. This means thinking through a plan, testing to minimize risks (ensuring the software and its functions are working as intended), focusing on quality, and then creating documentation from a process perspective.

However, pharmaceutical companies should be careful to consider that FDA guidance is still a draft when deciding how to apply it: It may change before the issue is in its final version. Furthermore, at this stage, it is unclear how many other different national authorities around the world will accept the CSA draft guidance. Pharmaceutical facilities that must concurrently comply with US FDA and other national authorities will need to balance to what extent they apply the CSA draft guidance. This will ensure that they do not inadvertently fail to adhere to the mix of different regulatory expectations that are applicable to their facilities.

The current draft of the FDA guidance applies to the medical device and biologics space, although the governing principles are expected to be agreed on and recommended by regulatory agencies also in the pharmaceutical space. Meanwhile, a number of regulated companies of all the previously mentioned regulated spaces have initiated the process of switching from a CSV repetitive strategy (where persons are used just to re-execute the validation approach established in the past) to a critical thinking approach. This transition should be carefully planned and deployed because it requires a substantial change in the mindset of all actors involved in the assurance activities related to a computer system.

References

1. US Food and Drug Administration. "Computer Software Assurance for Production and Quality System Software. Draft Guidance for Industry and Food and Drug Administration Staff." September 2022. https://www.fda.gov/media/161521/download
2. International Society for Pharmaceutical Engineering. *ISPE GAMP® Good Practice Guide: Enabling Innovation.* North Bethesda, MD: International Society for Pharmaceutical Engineering, 2021.
3. International Society for Pharmaceutical Engineering. *ISPE GAMP® 5 Guide: A Risk-Based Approach to Compliant GxP Computerized Systems (Second Edition).* North Bethesda, MD: International Society for Pharmaceutical Engineering, 2022.
4. US Food and Drug Administration. "CFR - Code of Federal Regulations Title 21, Annex 11." March 1997. www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/CFRSearch.cfm?CFRPart=11
5. US Food and Drug Administration. "Part 11, Electronic Records; Electronic Signatures - Scope and Application, Guidance for Industry." September 2003. www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application

About the authors

Gaurav Walia is Senior Associate Partner and a Principal Subject Matter Expert of Computer Systems Validation, CSA, and Data Integrity with PQE Group. He has more than 26 years of experience in pharmaceutical and medical devices and consumer healthcare, with extensive experience in end-to-end processes such as CSA, data integrity, validation and qualification, and much more. He is an active speaker at industry conferences and has provided numerous training programs in CSV and data integrity. He is actively engaged with the Knowledge Exchange Network (KENX) and serves as Vice Chairman, Chicago Chapter, American Society for Quality (ASQ). He joined ISPE in 2022.

Danilo Neri, PhD, is Partner and Vice President Executive at PQE Group and Board Member at the Parenteral Drug Association (PDA) Italy Chapter. He is a Principal Consultant, with experience working in the pharmaceutical industry. Danilo is a certified PDA Auditor in Computer Products Supplier Auditing Process Model since 2003. He is recognized as an expert in the fields of data integrity assurance, CSV, and serialization. He speaks at seminars and events dedicated to CSV. He has over 20 years of experience in managing multi-site, international projects for manufacturing execution systems, enterprise resource planning, and serialization system validation. He is a member of the ISPE Italy Affiliate Board and a delegate for International Relationship. He graduated with a bachelor's degree in physics and a PhD in engineering. He joined ISPE in 2000.