Tricentis

From CSV to CSA: Shift to a risk-based approach to improve software quality and FDA compliance

INTRODUCTION

Computer system validation (CSV) approaches have deviated little since the FDA first released its guidance in 1997. Meanwhile, just about everything else about the way software is built, tested, and released has changed. When it comes to CSV, outdated is a bit of an understatement. That's not to say the FDA hasn't evolved its approach to compliance over the last 25 years.

THE EVOLUTION OF CSV

In 2003, the agency released updated guidance to clarify and expand on those 1997 "General Principles of Software Validation. This guidance introduced the concept of a risk-based approach, encouraging organizations to tailor the extent of their validation processes to the nature and intended use of the system.

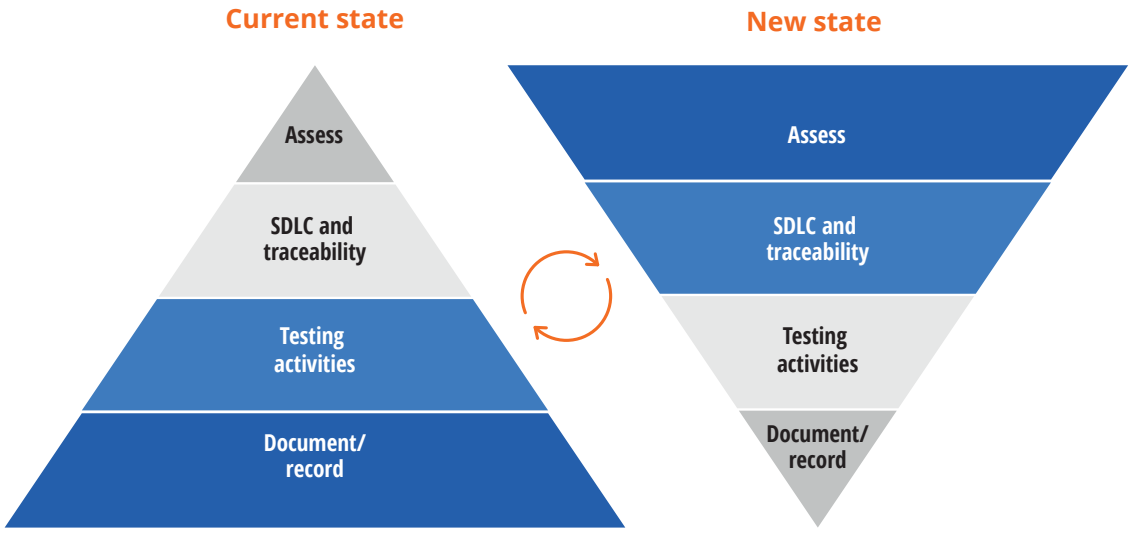
The intent was to reduce the documentation burden, but a fear of being noncompliant and a lack of clarity around the regulation meant many organizations continued treating everything as high-risk … and continued over-documenting, over-testing, feeling overwhelmed by the burden of it all. According to some interpretations, checking the boxes of compliance became more important than system quality.

Rather than achieving the original intent of promoting quality best practices and better aligning to rapidly modernizing software delivery methods, the FDA decided its 2003 guidance inhibited agile software practices. Its October 2011 report, developed as part of its Case for Quality initiative, found that the burden of CSV deterred investments in technology and that "companies believe(d) that the current regulatory framework slows process innovation around quality."

Its latest draft guidance, Computer Software Assurance for Production and Quality System Software," released in September 2022, further specifies how life sciences organizations can remain compliant while improving quality, removing non-value add activities, and focusing testing on high-risk areas.

To help manufacturers achieve CSV and ultimately FDA compliance, the computer software assurance (CSA) framework provides clarity on the stance and methodology used to determine what is high risk and what is not, thereby maximizing efficiency efforts by manufacturers. The clarification in the CSA approach flips the paradigm to focus on critical thinking (risk-based assessment), traceability of assurance needs, testing activities, and documentation – in that order.

CSV had previously emphasized an opposite approach in software validation, with the lion's share of attention devoted to documentation, and a descending focus on testing activities, assurance traceability across the SDLC, and critical thinking, as demonstrated in the graphic following.



Source: Gartner

Using a risk-based approach is hardly new – organizations such as the International Society for Pharmaceutical Engineering (ISPE), author of the GAMP® 5: A risk-based approach to compliant GxP computerized systems, revised in July 2022, have been evangelizing this concept for two decades – but for the many HLS organizations who maintain a traditional approach, it will require a significant shift in both culture and mindset.

So, what should organizations looking to transition from CSV to CSA know? And what should a risk-based approach look like within a CSA framework? Read on to find out.

BEST PRACTICES FOR IMPLEMENTING A RISK-BASED APPROACH

CSA consists of four major steps in its validation pathway:

Step 1. Identifying the intended use

To identify the software's intended use, ask: Is it part of the ***direct*** or the ***indirect*** software system?

Software designed for automating production processes, inspection, testing, or the collection and processing of production data is classified as part of the ***production system***, also described as a ***direct system***, due to its direct impact on product quality or patient safety. These systems are considered high-risk, accompanied by appropriately rigorous assurance activities. It is these systems that the updated CSA guidance focuses on.

Indirect systems include software designed to automate processes, data collection, or record maintenance of quality systems regulation, along with production or quality support software, and typically carry lower risk than direct systems due to lower risk for patient safety or product quality.

Step 2. Determining the risk-based approach

So, what should a risk-based approach to assurance look like within the context of CSA?

Many testing portfolios suffer from the same problem — a large volume of tests but little insight into how well the highest risk areas are covered. It's virtually impossible to validate every single component of your software system, especially with today's complex and highly integrated environments. This is exactly the challenge that a risk-based approach intends to address.

Complete a risk assessment

Rather than giving every requirement equal focus during testing, teams should instead assess the degree of risk associated with each requirement and then prescribe a correlative degree of focus during testing.

From GAMP® 5 we have a simple method for assessing risk in software systems:

- **Severity:** Impact on patient safety, product quality, data Integrity, or other harm
- **Probability:** Likelihood of that fault occurring
- **Risk class:** severity x probability

Factoring in one more characteristic, detectability, or the likelihood that fault will be detected before harm occurs, gives us the final assessment of risk.

If the system component is calculated to have a high risk priority, it should receive a correlatively high degree of critical thinking when planning its assurance activities. To assign a risk priority score, multiply risk class by detectability.

Determine your current level of risk coverage and address gaps

Now that you understand the highest-risk areas, you will need to assess your current test case library against those risks. This is a new way of looking at your test case library and is likely to shed light on areas where you need to add tests to establish acceptable coverage. If you're starting from scratch, plan to begin by creating test cases for your highest risks, then work back from there.

Once you start adding tests, linking them to requirements will help you identify the risk contribution of each test. Moreover, this correlation between tests, requirements, and risk is essential for obtaining risk-

based reporting. With everything linked and correlated, you'll gain insight into:

- Prominent gaps in your risk coverage
- The impact of your test failures
- The readiness of particular requirements
- The application's overall release readiness

Step 3: Determining the appropriate assurance activities

How to boost assurance activity efficiency with test automation and test management:

The FDA recommends two types of testing activities reflective of the software's risk level. For low-risk coverage areas, unscripted testing (testing actions not prescribed by written instructions) is the appropriate activity, and includes ad-hoc testing, error-guessing, and exploratory testing. All told, these are largely based on the tester's instinct and "rules of thumb."

However, for high-risk components, scripted testing becomes the appropriate assurance activity.

Scripted testing in this FDA context refers to dynamic testing in which the tester's actions are prescribed by written instructions in a test case, including robust and limited scripted testing. This high-risk coverage area with high-value scripted testing activities introduces the potential for accelerated adoption and scaling of CSA through robust risk-based test automation tools.

Testing the highest-risk areas, instead of testing everything, will undoubtedly save your team time. But implementing this approach is a significant shift in process for many teams. It will require setting aside time to complete a risk assessment and evaluate the current test case library. By automating rote manual tasks and freeing up more time for test-planning strategy, implementing or scaling automated testing can ease the transition from CSV to CSA.

Many organizations begin by automating their regression test suite so that it can be run before each release. This frees up testers' time to focus on new functionality that needs to be tested. If you are starting from scratch, you can apply a risk-based approach to your automation strategy – automating the most critical regression tests first. Test automation tools like Tricentis Tosca offer risk-based capabilities that will prioritize your test cases for you, based on the level of risk you assign to each requirement.

You can take it a step further by linking your requirements management system, like Jira, to your test cases and results with a test management tool like Tricentis Tosca. When requirements are linked, you can instantly gain insight on which critical tests passed or failed during execution.

The result? Significantly improved risk coverage with less time spent on testing.

Step 4. Establishing appropriate records

How to establish and maintain appropriate records:

After the assurance activities have been completed, it remains critically important to compile records of its successful assessment and approval, just as critical a step as originally defined in CSV. The difference lies in how much detail is recommended for each different assurance activity.

For all assurance activities, the FDA advises recording the intended use, risk determination, and assurance activities conducted.

Documentation of assurance activities should always include a pass/fail result for each test case, issues found and disposition, who performed which test when, and established review and approval when appropriate for all types of testing activities.

However, only robust scripted testing activities are recommended, including detailed reports of all testing activities performed (associated with the highest-risk direct software systems). For all other assurance activities, including limited scripting and unscripted testing activities, a brief summary is all that is advised, saving time in the documentation stage compared to former CSV practices.

To alleviate roadblocks in the electronic review and approval process of these assurance activities, a digital validation solution such as Tricentis Vera™ can accelerate critical approval, verification, and compliance management processes while still ensuring FDA compliance.

Unlike traditional e-signature and document-centric processes, Vera™ automates the validation process with configurable workflows, controls, and a data-driven approach that is designed for Agile and DevOps environments, including pre-execution approval of formally reviewed and approved automated Tricentis Tosca tests within qTest.

A stitch in time saves nine

By reducing efforts on testing and documenting everything, all the time, and to the same level of rigor, and applying that effort into critical thinking in test planning, organizations are actually able to increase quality. They are able to reduce errors in their critical systems by applying appropriate effort and resources where they should be focused, rather than being spread too thin with efforts being applied equally across

all systems, applications, and requirements – a virtually impossible task in today's fast-paced software development environment.

Further, while there can be some upfront cost and effort associated with planning for risk-based testing, once implemented, the savings in both cost and resource time begin to manifest quite substantially, and product quality improves. As Thomas Fuller once said in 1732, a stitch in time saves nine. Applied to the complex and careful world of Health/Life Sciences (HLS) validation, save the organization nine code errors, nine leaked defects, nine unvalidated patient data records, and take that stitch in your risk-based testing planning.

ABOUT TRICENTIS

Tricentis is a global leader in enterprise continuous testing. The Tricentis AI-based, continuous testing portfolio of products provide a new and fundamentally different way to perform software testing. An approach that's totally automated, fully codeless, and intelligently driven by AI. It addresses both agile development and complex enterprise apps, enabling enterprises to accelerate their digital transformation by dramatically increasing software release speed, reducing costs, and improving software quality. Widely credited for reinventing software testing for DevOps, cloud, and enterprise applications, Tricentis has been recognized as a leader by all major industry analysts, including Forrester, Gartner, and IDC. Tricentis has more than 2,500 customers, including the largest brands in the world, such as McKesson, Accenture, Nationwide Insurance, Allianz, Telstra, Dolby, and Vodafone.

To learn more, visit www.tricentis.com or visit one of our locations, www.tricentis.com/locations.

v. 0623