

An Introduction to Computer Software Assurance (CSA) for Pharmaceutical Labs



Computer software validation (CSV) is a well-established methodology for ensuring a software system does what it is intended to do. It has been around for decades, but in those intervening years, the technologies available to pharmaceutical operations, including in laboratories, have significantly advanced. As a result, the FDA has been working on a new, more modern, and more efficient approach known as computer software assurance (CSA).

In brief terms, the aim of CSA is to move away from the compliance and documentation focus of CSV, replacing it with [validation/assurance processes](#) that have a greater emphasis on best practices and process improvement. All this while [remaining compliant with GAMP 5](#). Let's unpack that a little bit more.

Risk-Based Approach Using Critical Thinking

It is widely believed the compliance-focused, documentation-heavy approach that is the hallmark of CSV doesn't add value. It can also be a barrier to new technology implementation, and it can stifle innovation. The CSV method of software validation is also costly, time-consuming, and resource intensive.

CSV does have its benefits and it has served the life sciences sector for decades. However, the FDA (along with the ISPE – International Society of Pharmaceutical Engineers) believes it is now time for software validation to get an update. It wants (and many in the industry want) to make assurance processes and objectives more aligned with the capabilities and realities of modern technologies.

What does this mean? Here are the main points:

- CSA involves taking a risk-based approach to software validation
- Critical thinking is at the core of the process
- Validation efforts for a software system should not be more than is necessary to address the level of risk

It's at the Draft Guidance Stage

The [FDA published draft guidance on CSA](#) in September 2022, so it is not finalised yet. That said, work on CSA has been ongoing for several years, and it represents the clear direction of travel for software validation in the life sciences sector. In fact, the FDA has advocated a risk-based approach to software assurance for some time now, with the new draft guidance bringing additional clarity. As a result, many organisations have already started the transition.

It's also important to point out that the draft guidance for CSA was developed primarily for medical device manufacturers. However, it is widely expected to also apply to other parts of the life sciences sector, including the pharmaceutical and biopharmaceutical industries.

It only applies to non-product software. This means enterprise-level platforms like ERP systems, as well as production and quality control solutions like MES and QMS systems. It also applies to software applications and platforms used in lab environments, such as LIMS. Implementing or [upgrading a system like Empower](#) is a specific example of where CSA can be applied.

What about the EU? While the CSA guidance is specific to the FDA, EU GMP Guide Annex 11 "Computerised Systems" also has a risk-based focus on software validation. Furthermore, the CSA guidance is part of the FDA's Case for Quality programme, and it has been collaborating with EU regulators and the UK's MHRA as part of this programme.

The Main Differences Between CSA and CSV

The CSV Approach

The CSV approach to software validation mostly involves the creation of documentation. So, for example, you buy a new software application and then set about recreating the documentation produced by the vendor. Testing is part of the process, but most of the time spent validating the software is taken up with producing documentation.

With customised off-the-shelf software, this often involves recreating documentation that has already been produced by the vendor. The result is a lot of duplication of effort. This duplication of effort doesn't exist with custom developed software. With this type of software, the traditional CSV approach to validation will still play an important role.

It's also important to consider the main aim of the CSV process. More often than not, it is to create the necessary documentary evidence that will be needed whenever an audit takes place.

As already mentioned, duplication of effort is one of the downsides of this approach. It is also inflexible, time-consuming, and open to human error. It also misses the main point of validating software, which is to improve the quality of computer systems and ensure they are fit for their intended use. Having the [necessary evidence for a regulatory audit](#) should be secondary.

CSA Explained

CSA involves taking a risk-based approach to software assurance. This means the highest level of validation is only required for software applications that directly impact patient safety and/or product quality. Where the risk level is assessed to be lower, the validation requirements are also lower.

What does this mean for the validation process? There are two main areas – documentation and testing.

In terms of documentation, there is no need to recreate documentation for customised off-the-shelf software if the vendor has already produced it, providing it is audited and deemed to be of a high enough quality to be deemed compliant ready. This point alone significantly decreases the validation burden.

In terms of testing, the assurance requirement depends on a risk analysis of the software. In general, software applications will fall into three main types of testing categories:

- Scripted testing – this is for applications that have a direct impact on safety or quality. The testing required for these types of applications is very similar to the testing that is traditionally performed when using the CSV approach.
- Unscripted testing – this is for software applications that have an indirect impact on quality and safety, so are deemed to be at a medium risk level. For these types of software applications, testing needs to be completed and documented with defined objectives and criteria for whether the app has passed or failed the test. Unlike scripted testing, step-by-step testing procedures don't need to be created or documented.
- Ad-hoc testing – this is for software applications that don't fall into either of the above two categories. In other words, it is for the lowest-risk applications. The process mostly involves exploratory testing.

The CSA Process in Summary

A summary of the CSA process to validate a software system can be described in four main steps:

- Establish the intended use of the software

- Determine the level of risk based on the impact of the software on product quality and patient safety
- Decide on and then implement the assurance activities, i.e., testing
- Create appropriate documentation, particularly in relation to testing, reusing vendor documentation where applicable

The Benefits of CSA Compared to CSV

- Increased flexibility
- Faster process
- Less resource-intensive, enabling a more efficient use of resources
- Eliminates duplication of effort and work
- More focused on ensuring software is safe and suitable for its intended use
- Facilitates different approaches based on risk, moving validation away from the inefficient one-size-fits-all approach
- Reduces levels of human error in the software testing part of the process

The Next Steps

As mentioned above, CSA represents the clear direction of travel for the FDA, but that doesn't mean CSV is being dropped completely. It still remains a valid approach for the assurance of software systems in pharmaceutical operations.

That said, there are compelling reasons for switching to the CSA approach, not least because of the business, process, and resource benefits.

The best next steps will depend on your organisation, lab, and project, as well as the [lab IT and software integration partner](#) that you work with, although CSA is definitely a validation process that is worth careful consideration.