Computer Software Assurance

- An Interpretation and Future

Center of Excellence in Regulatory Science and Innovation &
Masters of Science in Regulatory Science and Food Safety Regulation – 2023 Writing Competition

Pavithra Peranamallur Thirumalai

Word Count:2939 (wordcount.com; excluding references)

Title: Computer Software Assurance - An Interpretation and Future

Submitted By: Pavithra Peranamallur Thirumalai

Abstract:

21 CFR Part 11 was introduced in 1997 and defined the requirements for electronic signatures and electronic records. Six years later in 2003, FDA published the 'Guidance for Industry Part 11, Electronic Records; Electronic Signatures — Scope and Application'. The approach to Part 11 compliance is criticized to be document heavy and strenuous to companies. Emphasis is provided to documentation rather than critical thinking and a risk-based approach. To overcome this criticism, in 2022 FDA published a draft guidance titled 'Computer Software Assurance for Production and Quality System Software' and Computer Software Assurance became the most discussed term among CSV professionals. The new guidance provides more emphasis on critical thinking, followed by testing and documentation.

This article analyses the CSA approach, discusses its advantages, challenges in adoption and future.

Keywords: Computer Software Assurance (CSA), Computer Software Validation (CSV), Part 11, critical thinking.

Background

Under Title 21, Chapter I, Subchapter A and Part 11 of the Code of Federal Regulations[1], the FDA defined the statutory requirements for electronic records and signatures. Part 11 provides the concept of an open and closed system, validation of systems, audit trails including its retention period, differences between an electronic signature and a handwritten signature, requirements for an electronic signature and maintenance of an identification and passcode. At the time of its publication, 21 CFR Part 11 was well received because prior to 1997 there was no regulation governing the use of electronic records and signatures.

In 2002, FDA's CDRH published the 'General Principles of Software Validation; Final Guidance for Industry and FDA Staff'. This guideline described the concept of software validation and ways to perform validation. This guidance primarily pertains to the medical device industry[3] and its scope was not expanded to drug manufacturers.

In 2003, FDA published the 'Guidance for Industry Part 11, Electronic Records; Electronic Signatures — Scope and Application' which elaborated the approaches to validation, audit trail, use of legacy systems and record maintenance[2].

CSV – Convincing or Confusing?

FDA defines software validation as objective evidence that a software can consistently perform per user needs and its intended uses. While the definition of CSV is simple, industry experts opine that there is a gap between what the FDA requires and what the industry perceives.

Often CSV is viewed as testing and documentation by the industry and there is common practice to perform CSV towards the end of system implementation. CSV Team isn't involved in any developmental discussions or code verification activities and by the time CSV activities begin, there is Upper Management pressure to release the system for use. In many companies, CSV is

considered as an event rather than process. Code changes at the end of system implementation can be costly and difficult to implement. The lack of software engineering concepts is attributed as the main cause for the misconceptions[5] and companies either overdo validation or do not meet FDA requirements. CSV concepts such as traceability, audit trail, and source code review should be part of the software building process rather than an ancillary task towards the end. To overcome this drawback, the concept of 'quality-by-design' (QbD) was introduced in 2000s and the industry tried to move from 'quality-by-inspection' to 'quality-by-design'. The applicability of QbD is not prevalent amongst companies developing software for pharmaceutical, biopharma or medical device industries and the concept is more relevant to drug development process.

On the other hand, validation concept is never a one-size-fits-all since validation approach tend to change. A flexible, risk-based, case-by-case approach is taken based on system's intend use and the risk it carries. The word 'flexible' comes with its own problems since it cannot be defined. While the industry can build an approach for CSV, it isn't necessary that the FDA would accept the approach. FDA's audit findings related to code review, traceability, and documentation of requirements and test results is considered bureaucratic[4] implementation of regulations rather than ensuring quality of the software.

One other perspective is that FDA Inspectors themselves are not clear with FDA's expectations of computer system validation[5]. Many inspectors are not tech-savvy to understand the software developmental lifecycle or how it directly or indirectly affects product quality and compromising patient safety. In FDA audits, many inspectors perform a surface level audit on 21 CFR Part 11 requirements, and this could be the cause of lesser warning letters in this area.

The need for a new approach:

With the advent of newer technologies, FDA has been slow in pushing for a new regulation or updating existing regulations or providing a new guidance document. In recent times there is increased adoption of electronic systems and the aim to go paperless is greater than ever, however, we are still following regulations and guidance documents that were published 20 years ago. The ambiguity created by 21 CFR Part 11 and the related guidance documents led to the creation of the guidance document titled 'Computer Software Assurance for Production and Quality System Software' in 2022.

CSA Approach – A Brief:

The CSA guidance has been prepared by the Center for Devices and Radiological Health (CDRH) and the Center for Biologics Evaluation and Research (CBER) in consultation with the Center for Drug Evaluation and Research (CDER), Office of Combination Products (OCP), and Office of Regulatory Affairs (ORA). The guidance is applicable to medical device industry, drug manufacturers and any group within the pharmaceutical space that uses electronic records and signatures.

CSA is a risk-based approach to ensure software is built for its intended use and follows the least burdensome approach[6] wherein validation is performed to address risk only. It also emphasizes on maintaining the software in a validated state through the software' life.

CSA – The Framework:

The CSA framework[6] consists of identifying the intended use for the software, determining a risk-based approach and assurance activities, and creating appropriate records:

1. **Identifying the Intended Use**

   Per 21 CFR 820.70(i), any software that is used as part of the production or quality system needs to be validated for its intended use[7]. In general, software can be categorized as:

- Direct Impact - one being used for a production or quality system process.
- Indirect Impact - one that supports a production or quality system process.

The regulation applies to both categories, however, it is essential to thoroughly understand the features, functions, and operations of a software when determining the intended use. As the complexity of software increases, each feature, function, and operation can have different intended use(s) and can pose a different risk to product quality.

FDA recommends documenting the intended use as part of the company's Standard Operating Procedures (SOPs). Identifying the intended use of the software will help in determining a risk-based approach.

2. Determining the Risk Based Approach

In a risk-based approach, foreseeable software failures are identified, classified and appropriate assurance activities are determined. The popular way to perform this activity for medical devices would be to list down all probable failures and classify them based on impact and occurrence.

Example of a heat map used to perform risk assessment:

		Impact		
		High	Medium	Low
Occurrence	High	Intensive Check	Intensive Check	Standard Check
	Medium	Intensive Check	Standard Check	Minimal Check
	Low	Standard Check	Minimal Check	Minimal Check

In the CSA framework,

- Any failure that has the potential to compromise production or quality system[6] is termed as a process risk.
- Any failure that has the potential to harm patients[6] is termed as a medical device or product[8] risk. The CSA guidance specifically considers medical device or product risks caused due to a quality problem that compromises safety.

The CSA's risk-based approach suggests classifying software as either "high process risk" and "not high process risk".

- High process risk – When software does not perform as intended, it may result in a quality problem that may foreseeably compromise safety[6]. Examples include but not limited to –
  - A software that can measure, inspect, analyze acceptability of product without manual intervention.
  - A software that can monitor and maintain temperature, pressure, or humidity in a sterile room and failure of which can impact product quality drastically.
- Not high process risk – When software does not perform as intended, it may not result in a quality problem that may foreseeably compromise safety[6]. Companies can choose to further categorize these software as "moderate," "intermediate," or "low" to determine assurance activities; however, FDA will consider them under "Not high process risk" category only. Examples include but not limited to –
  - A software that is used to monitor and manage Corrective and Preventive Actions.
  - A software capable of printing batch records in a presentable format.

Once the intended use and risk is identified, the next step is to identify the assurance activity.

3. Determining the Appropriate Assurance Activities

Simply put, a 'high process risk' needs more objective evidence when compared to a 'not high process risk'.

For a high process risk, FDA suggests performing scripted testing.

- Scripted Testing – Testing involves writing step-by-step actions and documenting that the tester performed the actions along with objective evidence.

- Robust Scripted Testing: Testing the entire software to ensure new features or functionalities did not affect the existing working of the system.
- Limited Scripted Testing: A combination of unscripted testing and scripted testing.

For a 'not high process' risk, FDA suggests performing unscripted testing.

- Unscripted testing – Dynamic testing where tester does not follow step-by-step instructions and does not need to collect objective evidence. Tester needs to have adequate knowledge of the software features, business process flow and SOPs to perform unscripted testing.
  - Ad-hoc Testing: As the name suggests, testing the software feature in a random manner without following a test protocol or documenting evidence.
  - Error-guessing: Testing is based on the Tester's knowledge of past failures and failure modes[6].
  - Exploratory Testing: Tester uses own experience to design and execute tests. It involves testing hidden parameters and hypothetical situations.

It is interesting to note that –

- A combination of unscripted and scripted testing can be used based on the risk of the software towards patient safety.
- For software that have an indirect impact on the production or the quality system, vendor validation documentation, software installation and configuration documentation can be leveraged. Scripted or unscripted testing is not necessary because, when software having direct impact are validated, it inherently validates the supporting software.
- The rigor of the assurance activities for the software feature can be reduced if additional measures are in place that ensure product or process quality. For example,

a software can design labels and a manual verification is performed before printing the labels. Here, the rigor can be less since there is a manual intervention.

4. Establishing the Appropriate Record

When something is not documented, it is not done. FDA needs objective evidence to prove that the software feature, function, or operation was assessed and performs as intended[6]. The following elements needs to be documented:

- The intended use of the software
- Risk-Based Analysis – The approach and results
- Test Versions
- Test Type (scripted / unscripted)
- Test Aim and Activities (test steps, test results, tester details)
- Issues
- Summary

As the least burdensome approach, FDA encourages the industry to use system-generated objective evidence such as system logs, audit trails rather than the traditional paper-based test scripts and screenshot evidence.

Application of the Least Burdensome Approach:

CSA focuses on being test oriented than documentation oriented. The below heat map is an example and created after understanding the concepts of intended use, risk-based approach, and appropriate assurance activities.

		Intended Use	
		Direct	Indirect
Risk	High Process Risk	Scripted Testing	Combination of Scripted and Unscripted Testing
	Not High Process Risk	Unscripted Testing	Leverage vendor documentation or software installation or configuration

The heat map indicates that scripted testing is not always necessary. Unscripted testing or leveraging vendor documentation is acceptable if risks are properly assessed.

CSA – The Advantages:

The CSA approach steers companies from a focusing on satisfying compliance goals to ensuring software quality. CSV Team will not be burdened to create the same set of documents for all projects, but rather apply critical thinking skills and tailor document packages according to software risks.

Other advantages include -

1. Applying CSA concept does not need a statutory change, but clearly defines FDA's current thinking of software validation concepts[9].
2. CSA concept is supported by International Society for Pharmaceutical Engineering (ISPE)[9, 10].
3. The guidance provides a clear distinction between software that have a direct, indirect and no impact on the production or quality systems. This differentiation is crucial for the industry since many companies were following an 'one-size-fits-all' method and it led to creation of unwanted documentation.

4. The risk-based approach is based on impact to patient safety and product quality and is assessed based on the software's risk[9]. The concept of classifying software into 'high process risk' and 'not high process risk' is simple and easy to understand.
5. While unscripted testing is not a new to the industry, the CSA approach formalizes the concept and provides assurance that it is an acceptable form of assurance activity. CSV personnel will be able to focus on testing rather than documentation. The time used to create test scripts, review and approval of the test scripts, execution and documentation of the test scripts will be reduced up to 80%[9] due to formalizing the unscripted testing.
6. Exploratory and error-guessing testing is based on the tester's previous knowledge of the system and ensures failures are tested multiple times before release into the production. The probability of encountering failures in production is less when it is tested multiple times in the test environment.
7. Emphasis on performing vendor audits. This can be helpful to assess if software is capable of handling functionalities for a pharmaceutical or biopharma or medical device company even before purchasing the software, leading to cost savings (why buy a software if it isn't compliant?)
8. Leveraging vendor documentation for indirect impact software is an absolute necessity. This is mainly because such software is purchased and used as-is without changes to its configurations. Performing any CSV activities on such software are repetitive and does not uncover any new issues.

CSA – The Challenges:

While CSA is the new kid in the block, it does not replace CSV. The CSA guidance is a supplement to the Section 6 of the "General Principles of Software Validation" Guidance. In future both CSA and CSA will coexist and may cause confusion[11].

- Regulatory Contradictions[11]:
    - The guideline specifies that infrastructure supporting software like servers, databases and peripherals do not require any qualification or documentation. Per EU GMP Annex 11, IT infrastructure needs to be qualified[12]. Ensuring IT infrastructure quality need not be as stringent as EU Annex but as lenient as the CSA approach - a middle ground needs to be identified.
    - Per EU GMP Annex 11 Section 4.4, requirements need to be traceable through the lifecycle of the software[12]. This is the reason for the 'V' model of validation[13], where unit testing traces to module design, integration testing traces to architecture design, system testing traces to system design and user acceptance testing traces to requirement design. The 'V' model ensures there is traceability through the lifecycle of the product development. CSA guidance mentions that traceability is required only for scripted testing and not required for unscripted testing. This is contradicting to EU GMP Annex 11.
- Unscripted testing is not undocumented testing:
    - When CSA introduced the term 'unscripted testing' and the least burdensome approach, many interpretations dubbed it as undocumented testing. Per the CSA guidance, unscripted testing requires the following documentation[6]:
        - Intended use.
        - Risk determination.
        - Summary description of failure-modes tested, and testing performed.
        - Issues found and disposition.
        - Conclusion statement.
        - Record of who performed testing and date of testing.
        - Established review and approval when appropriate.

The above list is exhaustive and equal to the documentation created for a scripted testing. The main difference between unscripted and scripted testing is unscripted testing does not require a test plan and pass/ fail noted for every test case. While this is less documentation, it cannot be concluded as an undocumented testing. Clarifying the term 'unscripted' testing can be helpful.

- Issues with error-guessing and exploratory unscripted testing:
    - People attrition is an issue for any Project Team and the CSV Team is no different. For error-guessing and exploratory unscripted testing, testers need to have a prior working knowledge of the software and a history of the failures. A tester can perform complete testing only when they have a good command on the ever-changing business process. With a CSV personnel departing the department, this knowledge is lost, and error-guessing and exploratory unscripted testing becomes a challenge for the newcomer. Also, this testing cannot be performed for a new software since we cannot predict past failures of a new software. In such cases, companies will be forced to perform scripted testing only. Unscripted testing is more people-based whereas scripts testing is process-based. People-based systems are risker for companies to manage than process-based systems.
- Applicability of CSA to different software types used in different settings:
    - The examples mentioned in the Appendix A of the guidance is primarily focused on Commercial-Off-The-Shelf (COTS) software. The CSA guidance lacks examples of software used in a laboratory settings like Process Analytical Technology (PAT)[11]. Per GAMP, there are other software categories such as infrastructure software, non-configured software, and custom software. FDA can consider providing examples for all different software to resonate with different set of audience.

CSA – The Future:

The CSA approach provides deeper insight on how companies can manage risk. Flushing out the types of testing along with the documentation requirements for each type of testing provides the much-needed clarity and leaves less room for guess work. The examples mentioned in the document, though not wholesome, can be useful for beginners and seasoned CSV personnel alike.

Though the guidance is a good starting point, many industry experts opine that the CSA concept is abstract and not relatable. Maybe the industry expected a step-by-step or a how-to kind of guidance document for the CSA concept. Commonly used CSV terms like user requirements, installation, operational and performance qualification (IQ/OQ/OQ) are not mentioned in the guidance, hence, the adoption from CSV to CSA can take more time.

The pharmaceutical industry is slow to adopt changes and is using the CSV concept for 20 years now. Only time will tell if the new CSA guideline was able to resolve all ambiguities related to 21 CFR Part 11 or in the process created newer ambiguities.

References:

1. 21 CFR Part 11; https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11
2. FDA; Guidance for Industry Part 11, Electronic Records; Electronic Signatures — Scope and Application; 2003; https://www.fda.gov/media/75414/download
3. FDA; General Principles of Software Validation; 2002; https://www.fda.gov/media/73141/download
4. Gordon B. Richman; The Part 11 Controversy A Root Cause Analysis and Science-Based Solutions; Pharmaceutical Technology-06-01-2005, Volume 205 Supplement, Issue 3; 2005; https://www.pharmtech.com/view/part-11-controversy-root-cause-analysis-and-science-based-solutions-gordon-b-richman
5. Yin, T. (2010). Dissecting the 21 CFR part 11 controversy. Journal of Validation Technology, 16(1), 91-96. Retrieved from https://www.proquest.com/scholarly-journals/dissecting-21-cfr-part-11-controversy/docview/205475211/se-2
6. FDA; Computer Software Assurance for Production and Quality System Software - Draft Guidance for Industry and Food and Drug Administration Staff; 2022; https://www.fda.gov/media/161521/download
7. 21 CFR 820.70(i); https://www.ecfr.gov/current/title-21/chapter-I/subchapter-H/part-820/subpart-G/section-820.70#p-820.70(i)
8. Compliance Group; CSA | FDA | Computer Software Assurance Risk-Based Approach [YouTube Video]; 2022; https://www.youtube.com/watch?v=W_Myw73uaHM
9. Darren Geaney; What is Computer Software Assurance (CSA) and why are the FDA transitioning from traditional Computer System Validation?; 2020; https://kneat.com/article/what-is-computer-software-assurance-csa-and-why-are-the-fda-transitioning-from-traditional-computer-system-validation/

10. Charlie C. Wakeham; Why We All Need Critical Thinking; 2022; https://ispe.org/pharmaceutical-engineering/ispeak/why-we-all-need-critical-thinking
11. R.D. McDowall; CSA: Much Ado About Nothing?; Spectroscopy, April 2023, Volume 38, Issue 4; Pages: 7–13,34; https://doi.org/10.56530/spectroscopy.hm6969t6
12. EU GMP Annex 11; https://health.ec.europa.eu/system/files/2016-11/annex11_01-2011_en_0.pdf
13. ProfessionalQA.com; V Model; 2019; https://www.professionalqa.com/v-model