ISPE®

Draft Guidance: **FDA-2022-D-0795 Computer Software Assurance for Production and Quality System Software**

Comments submitted by the International Society for Pharmaceutical Engineering (ISPE), regulatorycomments@ispe.org

GENERAL COMMENTS ON THE DOCUMENT
ISPE welcomes the opportunity offered by FDA for a public consultation on Computer Software Assurance (CSA) for Production and Quality System Software draft guidance. ISPE welcomes that this CSA guidance focuses industry computerized systems management effort on quality (i.e., fitness for intended use throughout the design and operational life cycle) using a risk-based framework for computer software assurance throughout the software's lifecycle.
There has been considerable feedback from many ISPE members relating to the narrow scope of the guidance i.e., production and quality systems for medical devices. ISPE sees a positive link between the CSA guidance superseding Section 6 of the General Principles of Software Validation, and the referencing of those same General Principles of Software Validation within the Part 11, Electronic Records; Electronic Signatures – Scope and Application guidance, which is applicable across all industries regulated by the Food, Drug and Cosmetic Act. ISPE understands that FDA accepts the use of any effective methodology for the validation and management of computerized systems, but some level of public affirmation that, for example, CDER and ORA will accept CSA approaches would provide assurance to the industry and increase the adoption and acceptance of CSA guidance.
ISPE suggests that Quality Risk Management terminology and principles as described in ICH Q9 (Revision) are used throughout. The use of ICH terminology and principles should lead to more consistent interpretation by industry and regulators and facilitate understanding and potential acceptance by other regulatory agencies.
Spectrum of Risk and Intended Use: Based on the rationale given below, and to harmonize with EU GMP, Annex 11, Computerised Systems (and industry guidance, e.g., ISPE GAMP 5) it is recommended that final guidance should regard system lifecycle tools as not considered to be used as part of production or the quality system and, therefore, not validated under 21 CFR 820.70(i). Life cycle tools include "Software intended for use

GENERAL COMMENTS ON THE DOCUMENT

as development tools that test or monitor software systems or that automate testing activities for the software used as part of production or the quality system, such as those used for developing and running scripts"

The draft guidance adopts an approach based on a spectrum or continuum of risk, very similar to GAMP 5. Lines 292 -295: "FDA acknowledges that process risks associated with software used as part of production or the quality system are on a spectrum, ranging from high risk to low risk. Manufacturers should determine the risk of each software feature, function, or operation as the risk falls on that spectrum, depending on the intended use of the software."

ISPE welcomes that FDA is presenting the process risks in a binary manner, "high process risk" and "not high process risk" since this aligns well with industry guidance (i.e., GAMP) philosophy of "focus on critical aspects".

The detailed approach to arranging or classifying systems along the spectrum based on intended use differs to industry guidance (e.g., GAMP 5) approach in a subtle way. The guidance notes (lines 151 – 182) that regulation requires manufacturers to validate software that is used as part of production or the quality system for its intended use (see 21 CFR 820.70(i)),

- Software considered to be used directly as part of production or the quality system
- Software considered to be used to support production or the quality system:

and it distinguishes the software above with:

- Software not considered to be used as part of production or the quality system which does not have to be validated under 21CFR 820.70(i).

The second "supporting" category currently includes in lines 163 - 165: "Software intended for use as development tools that test or monitor software systems or that automate testing activities for the software used as part of production or the quality system, such as those used for developing and running scripts". The text in lines 170 – 172 and 484 – 485 go on to note that supporting software often carries lower risk, such that "the assurance effort may generally be reduced accordingly."

Applying the concept of "validation" to such tools raises a potential barrier and discouragement to their use, as well as potentially increasing cost without additional quality and safety benefits, depending on the interpretation of "validation" in the regulated company.

GENERAL COMMENTS ON THE DOCUMENT

This diverges from the GAMP 5 approach where testing tools are considered as Category 1 (and not GxP regulated). It also diverges from EU, Annex 11 which requires that “Automated testing tools and test environments should have documented assessments for their adequacy” but does not require validation.

The Guidance does, however, also note (lines 485 – 490) that because assurance activities used “directly” in production or the quality system often inherently cover the performance of supporting software, assurance that this supporting software performs as intended may be sufficiently established by leveraging vendor validation records, software installation, or software configuration, such that additional assurance activities (e.g., scripted or unscripted testing) may be unnecessary.

Also, further, the guidance specifically notes (lines 477 – 479) that Computer System Validation tools such as automated testing tools are examples of additional controls or mechanisms that increase the level of assurance of software performance and could reduce effort which may be devoted to other forms of assurance activities.

The two paragraphs above suggest to ISPE that lifecycle tools may not be considered to be used as part of production or the quality system.

We suggest, for clarity, to include some text (e.g., in a footnote) to explain more some terms since there were comments from our membership that these terms require more explanation to assist with practical implementation.

- Unscripted tests: During unscripted testing, the tester may select the most appropriate and effective methods to test the software, based on their knowledge, training, and experience, to achieve system objectives and requirements, and fitness for intended use, and to identify and remove defects. The applied test methods will be based on the knowledge and experience of the software development and test team. Unscripted testing is a common and well-established software engineering technique, which is extremely effective in defect prevention. Test plans and/or test cases defining objectives, pass/fail criteria and documentation requirements should be produced. Unscripted testing is not undocumented testing.

  Unscripted testing may be used to test lower risk (i.e., not high process risk) features of a system. The selection of appropriate test methods, should not, however, be based solely on risk, but should be based on the objective of the test (e.g., defect identification), the nature of the component being tested, and other technical system and process characteristics.

GENERAL COMMENTS ON THE DOCUMENT
• Manufacturer: should clarify that this refers to the software user - some readers might take this to mean the software provider rather than the person using the software.
We suggest referencing IEC/ISO 29119(2022) as the latest revision.

Specific Comments on the Text

ISPE indicates text proposed for deletion with ~~strikethrough~~ and text proposed for addition with **<u>bold and underlining</u>**.

Section or Line Number	Current Text	Proposed Change	Rationale or Comment
Lines 43 through 57	FDA envisions a future state where the medical device ecosystem is inherently focused on device features and manufacturing practices that promote product quality and patient safety. processing systems used as part of production or the quality system.	We suggest dividing this chapter in 2 subparts • A) line 43-74 for Medical device • B) line 76-92 for software ~~computerized~~	This proposed organization will help with a more user-friendly format separating the requirements for medical devices and Software.
238ff	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.	Suggested to make the terms and thereby the difference clearer – or remove the term 'likely'.	The meaning of 'reasonably foreseeable' as opposed to 'likely' is not clear.
Lines 434 - 442	Risk-based assurance. The draft states: "In general, FDA recommends that manufacturers apply principles of risk -based testing in which the management, selection,	Please consider adding a statement(s) to clarify that, as well as risk, other aspects such what is being tested, by whom, at what stage of the	This model does not reflect the realities of software development and testing. An effective testing and assurance strategy cannot be defined based solely

Section or Line Number	Current Text	Proposed Change	Rationale or Comment
	prioritization, and use of testing activities and resources are consciously based on corresponding types and levels of analyzed risk to determine the appropriate activities. For high-risk software features, functions, and operations, manufacturers may choose to consider more rigor such as the use of scripted testing or limited scripted testing, as appropriate, when determining their assurance activities. In contrast, for software features, functions, and operations that are not high-risk, manufacturers may consider using unscripted testing methods such as ad-hoc testing, error-guessing, exploratory testing, or a combination of methods that is suitable for the risk of the intended use.	life cycle, in which environment, and for what purpose and objectives should be considered in detail in order to select appropriate and effective assurance approaches.	on risk priority, and must consider factors including: • The nature, technical aspects and the architecture of the component to be tested. • Who is performing the testing (developer, testing specialist, implementation team, process subject matter expert, end user). • Where in the software life cycle the testing is being performed. • The type of testing (structural, functional, static, performance, acceptance…etc.) and what the main objective is at that stage (e.g., removal of defects, vs detailed functional tests, vs challenging risk management controls, vs regression vs acceptance against requirements) This is very well discussed in the ISPE GAMP RDI Data Integrity by Design Good Practice Guide Appendix S2 on Computer Software Assurance, Section 19.3 Risk-Based Assurance.
Lines 444-447	For example, as part of a comprehensive assurance approach, manufacturers can leverage the following to reduce the effort of additional assurance activities	While the additional controls provide further assurance, we suggest encouraging the assurance of effective technical controls where possible and available.	This could be misread as suggesting a lower focus on the assurance of technical controls where there are also manual or procedural controls.

Section or Line Number	Current Text	Proposed Change	Rationale or Comment
Lines 477 – 479	The use of Computer System Validation tools (e.g., bug tracker, automated testing) for the assurance of software used in production or as part of the quality system whenever possible.	ISPE suggests replacing the term "Computer System Validation Tools" with "Computer System Life Cycle Tools"	The proposed phrase is more general, and more accurately describes the examples given.