Contains Nonbinding Recommendations

Draft – Not for Implementation

Computer Software Assurance for Production and Quality System Software

---

Draft Guidance for Industry and Food and Drug Administration Staff

DRAFT GUIDANCE

This draft guidance document is being distributed for comment purposes only.

Document issued on September 13, 2022.

You should submit comments and suggestions regarding this draft document within 60 days of publication in the *Federal Register* of the notice announcing the availability of the draft guidance. Submit electronic comments to https://www.regulations.gov. Submit written comments to the Dockets Management Staff, Food and Drug Administration, 5630 Fishers Lane, Room 1061, (HFA-305), Rockville, MD 20852. Identify all comments with the docket number listed in the notice of availability that publishes in the *Federal Register*.

For questions about this document regarding CDRH-regulated devices, contact the Compliance and Quality Staff at 301-796-5577 or by email at CaseforQuality@fda.hhs.gov. For questions about this document regarding CBER-regulated devices, contact the Office of Communication, Outreach, and Development (OCOD) at 1-800-835-4709 or 240-402-8010, or by email at ocod@fda.hhs.gov.

U.S. Department of Health and Human Services
Food and Drug Administration
Center for Devices and Radiological Health
Center for Biologics Evaluation and Research

Preface

Additional Copies

CDRH

Additional copies are available from the Internet. You may also send an email request to CDRH-Guidance@fda.hhs.gov to receive a copy of the guidance. Please include the document number 17045 and complete title of the guidance in the request.

CBER

Additional copies are available from the Center for Biologics Evaluation and Research (CBER), Office of Communication, Outreach, and Development (OCOD), 10903 New Hampshire Ave., Bldg. 71, Room 3128, Silver Spring, MD 20993-0002, or by calling 1-800-835-4709 or 240-402-8010, by email, ocod@fda.hhs.gov or from the Internet at https://www.fda.gov/vaccines-blood-biologics/guidance-compliance-regulatory-information-biologics/biologics-guidances.

Table of Contents

I. Introduction 4
II. Background 5
III. Scope 6
IV. Computer Software Assurance 6
V. Computer Software Assurance Risk Framework 7
A. Identifying the Intended Use 7
B. Determining the Risk-Based Approach 9
C. Determining the Appropriate Assurance Activities 13
D. Establishing the Appropriate Record 16
Appendix A. Examples 20
Example 1: Nonconformance Management System 20
Example 2: Learning Management System (LMS) 23
Example 3: Business Intelligence Applications 24

Computer Software Assurance for Production and Quality System Software

Draft Guidance for Industry and Food and Drug Administration Staff

This draft guidance, when finalized, will represent the current thinking of the Food and Drug Administration (FDA or Agency) on this topic. It does not establish any rights for any person and is not binding on FDA or the public. You can use an alternative approach if it satisfies the requirements of the applicable statutes and regulations. To discuss an alternative approach, contact the FDA staff or Office responsible for this guidance as listed on the title page.

I. Introduction[1]

FDA is issuing this draft guidance to provide recommendations on computer software assurance for computers and automated data processing systems used as part of medical device production or the quality system. This draft guidance is intended to:

- Describe "computer software assurance" as a risk-based approach to establish confidence in the automation used for production or quality systems, and identify where additional rigor may be appropriate; and

- Describe various methods and testing activities that may be applied to establish computer software assurance and provide objective evidence to fulfill regulatory requirements, such as computer software validation requirements in 21 CFR part 820 (Part 820).

When final, this guidance will supplement FDA's guidance, "General Principles of Software Validation" ("Software Validation guidance")[2] except this guidance will supersede Section 6 ("Validation of Automated Process Equipment and Quality System Software") of the Software Validation guidance.

[1] This 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).

[2] Available at https://www.fda.gov/regulatory-information/search-fda-guidance-documents/general-principles-software-validation.

For the current edition of the FDA-recognized consensus standard referenced in this document, see the FDA Recognized Consensus Standards Database.[3]

In general, FDA's guidance documents do not establish legally enforceable responsibilities. Instead, guidances describe the Agency's current thinking on a topic and should be viewed only as recommendations, unless specific regulatory or statutory requirements are cited. The use of the word *should* in Agency guidances means that something is suggested or recommended, but not required.

II. Background

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. FDA has sought to identify and promote successful manufacturing practices and help device manufacturers raise their manufacturing quality level. In doing so, one goal is to help manufacturers produce high-quality medical devices that align with the laws and regulations implemented by FDA. Compliance with the Quality System regulation, Part 820, is required for manufacturers of finished medical devices to the extent they engage in operations to which Part 820 applies. The Quality System regulation includes requirements for medical device manufacturers to develop, conduct, control, and monitor production processes to ensure that a device conforms to its specifications (21 CFR 820.70, Production and Process Controls), including requirements for manufacturers to validate computer software used as part of production or the quality system for its intended use (see 21 CFR 820.70(i)).[4] Recommending best practices should promote product quality and patient safety, and correlate to higher-quality outcomes. This draft guidance addresses practices relating to computers and automated data processing systems used as part of production or the quality system.

In recent years, advances in manufacturing technologies, including the adoption of automation, robotics, simulation, and other digital capabilities, have allowed manufacturers to reduce sources of error, optimize resources, and reduce patient risk. FDA recognizes the potential for these technologies to provide significant benefits for enhancing the quality, availability, and safety of medical devices, and has undertaken several efforts to help foster the adoption and use of such technologies.

Specifically, FDA has engaged with stakeholders via the Medical Device Innovation Consortium (MDIC), site visits to medical device manufacturers, and benchmarking efforts with other industries (e.g., automotive, consumer electronics) to keep abreast of the latest technologies and to better understand stakeholders' challenges and opportunities for further advancement. As part of these ongoing efforts, medical device manufacturers have expressed a desire for greater clarity regarding the Agency's expectations for software validation for computers and automated data processing systems used as part of production or the quality system. Given the rapidly changing

[3] Available at https://www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfStandards/search.cfm.

[4] This guidance discusses the "intended use" of computer software used as part of production or the quality system (see 21 CFR 820.70(i)), which is different from the intended use of the device itself (see 21 CFR 801.4).

nature of software, manufacturers have also expressed a desire for a more iterative, agile approach for validation of computer software used as part of production or the quality system.

Traditionally, software validation has often been accomplished via software testing and other verification activities conducted at each stage of the software development lifecycle. However, as explained in FDA's Software Validation guidance, software testing alone is often insufficient to establish confidence that the software is fit for its intended use. Instead, the Software Validation guidance recommends that "software quality assurance" focus on preventing the introduction of defects into the software development process, and it encourages use of a risk-based approach for establishing confidence that software is fit for its intended use.

FDA believes that applying a risk-based approach to computer software used as part of production or the quality system would better focus manufacturers' assurance activities to help ensure product quality while helping to fulfill the validation requirements of 21 CFR 820.70(i). For these reasons, FDA is now providing recommendations on computer software assurance for computers and automated data processing systems used as part of medical device production or the quality system. FDA believes that these recommendations will help foster the adoption and use of innovative technologies that promote patient access to high-quality medical devices and help manufacturers to keep pace with the dynamic, rapidly changing technology landscape, while promoting compliance with laws and regulations implemented by FDA.

III. Scope

When final, this guidance is intended to provide recommendations regarding computer software assurance for computers or automated data processing systems used as part of production or the quality system.

This guidance is not intended to provide a complete description of all software validation principles. FDA has previously outlined principles for software validation, including managing changes as part of the software lifecycle, in FDA's Software Validation guidance. This guidance applies the risk-based approach to software validation discussed in the Software Validation guidance to production or quality system software. This guidance additionally discusses specific risk considerations, acceptable testing methods, and efficient generation of objective evidence for production or quality system software.

This guidance does not provide recommendations for the design verification or validation requirements specified in 21 CFR 820.30 when applied to software in a medical device (SiMD) or software as a medical device (SaMD). For more information regarding FDA's recommendations for design verification or validation of SiMD or SaMD, see the Software Validation guidance.

IV. Computer Software Assurance

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.

In addition, computer software assurance establishes and maintains that the software used in production or the quality system is in a state of control throughout its lifecycle ("validated state"). This is important because manufacturers increasingly rely on computers and automated processing systems to monitor and operate production, alert responsible personnel, and transfer and analyze production data, among other uses. By allowing 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 computer software assurance approach provides flexibility and agility in helping to assure that the software maintains a validated state consistent with 21 CFR 820.70(i).

Software that is fit for its intended use and that maintains a validated state should perform as intended, helping to ensure that finished devices will be safe and effective and in compliance with regulatory requirements (see 21 CFR 820.1(a)(1)). Section V below outlines a risk-based framework for computer software assurance.

V. Computer Software Assurance Risk Framework

The following approach is intended to help manufacturers establish a risk-based framework for computer software assurance throughout the software's lifecycle. Examples of applying this risk framework to various computer software assurance situations are provided in **Appendix A.**

A. Identifying the Intended Use

The 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)). To determine whether the requirement for validation applies, manufacturers must first determine whether the software is intended for use as part of production or the quality system.

In general, software used as part of production or the quality system falls into one of two categories: software that is used directly as part of production or the quality system, and software that supports production or the quality system.

Software with the following intended uses are considered to be used **directly** as part of production or the quality system:

- Software intended for automating production processes, inspection, testing, or the collection and processing of production data; and
- Software intended for automating quality system processes, collection and processing of quality system data, or maintaining a quality record established under the Quality System regulation.

Software with the following intended uses are considered to be used to **support** production or the quality system:

- 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; and
- Software intended for automating general record-keeping that is not part of the quality record.

Both kinds of software are used as "part of" production or the quality system and must be validated under 21 CFR 820.70(i). However, as further discussed below, supporting software often carries lower risk, such that under a risk-based computer software assurance approach, the effort of validation may be reduced accordingly without compromising safety.

On the other hand, software with the following intended uses generally **are not** considered to be used as part of production or the quality system, such that the requirement for validation in 21 CFR 820.70(i) would not apply:

- Software intended for management of general business processes or operations, such as email or accounting applications; and
- Software intended for establishing or supporting infrastructure not specific to production or the quality system, such as networking or continuity of operations.

FDA recognizes that software used in production or the quality system is often complex and comprised of several features, functions, and operations;[5] software may have one or more intended uses depending on the individual features, functions, and operations of that software. In cases where the individual features, functions, and operations have different roles within production or the quality system, they may present different risks with different levels of validation effort. FDA recommends that manufacturers examine the intended uses of the individual features, functions, and operations to facilitate development of a risk-based assurance strategy. Manufacturers may decide to conduct different assurance activities for individual features, functions, or operations.

For example, a commercial off-the-shelf (COTS) spreadsheet software may be comprised of various functions with different intended uses. When utilizing the basic input functions of the COTS spreadsheet software for an 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. 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

[5] That is, software is often an integration of "features," that are used together to perform a "function" that provides a desired outcome. Several functions of the software may, in turn, be applied together in an "operation" to perform practical work in a process. For the purposes of this guidance, a "function" refers to a "software function" and is not to be confused with a "device function."

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 maintains a validated state. However, if a manufacturer utilizes 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.

For the purposes of this guidance, we describe and recommend a computer software assurance framework by examining the intended uses of the individual features, functions, or operations of the software. However, in simple cases where software only has one intended use (e.g., if all of the features, functions, and operations within the software share the same intended use), manufacturers may not find it helpful to examine each feature, function, and operation individually. In such cases, manufacturers may develop a risk-based approach and consider assurance activities based on the intended use of the software overall.

FDA recommends that manufacturers document their decision-making process for determining whether a software feature, function, or operation is intended for use as part of production or the quality system in their Standard Operating Procedures (SOPs).

B. Determining the Risk-Based Approach

Once a manufacturer has determined that a software feature, function, or operation is intended for use as part of production or the quality system, FDA recommends using a risk-based analysis **to determine appropriate assurance activities.** Broadly, this risk-based approach entails systematically identifying reasonably foreseeable 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, as applicable.

Note that conducting a risk-based analysis for computer software assurance for production or quality system software is distinct from performing a risk analysis for a medical device as described in ISO 14971:2019 – *Medical devices – Application of risk management to medical devices*. Unlike the risks contemplated in ISO 14971:2019 for analysis (medical device risks), failures of the production or the quality system software to perform as intended do not occur in a probabilistic manner where an assessment for the likelihood of occurrence for a particular risk could be estimated based on historical data or modeling.

Instead, the risk-based analysis for production or quality system software considers those 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. This guidance discusses both *process risks* and *medical device risks*. A process risk refers to the potential to compromise production or the quality system. A medical device risk refers to the potential for a device to harm the patient or user. When discussing medical device

risks, this guidance focuses on the medical device risk resulting from a quality problem that compromises safety.

Specifically, FDA considers a software feature, function, or operation to pose a high **process** risk **when its failure to perform as intended may result in a quality problem that foreseeably compromises safety, meaning an increased medical device risk.** This process risk identification step focuses only on the process, as opposed to the medical device risk posed to the patient or user. Examples of software features, functions, or operations that are generally **high process risk** are those that:

- maintain process parameters (e.g., temperature, pressure, or humidity) that affect the physical properties of product or manufacturing processes that are identified as essential to device safety or quality;
- measure, inspect, analyze and/or determine acceptability of product or process with limited or no additional human awareness or review;
- perform process corrections or adjustments of process parameters based on data monitoring or automated feedback from other process steps without additional human awareness or review;
- produce directions for use or other labeling provided to patients and users that are necessary for safe operation of the medical device; and/or
- automate surveillance, trending, or tracking of data that the manufacturer identifies as essential to device safety and quality.

In contrast, FDA considers a software feature, function, or operation not to pose a high process risk **when its failure to perform as intended would not result in a quality problem that foreseeably compromises safety**. This includes situations **where failure to perform as intended would not result in a quality problem,** as well as situations **where failure to perform as intended may result in a quality problem that does not foreseeably lead to compromised safety**. Examples of software features, functions, or operations that generally are **not high process risk** include those that:

- collect and record data from the process for monitoring and review purposes that do not have a direct impact on production or process performance;
- are used as part the quality system for Corrective and Preventive Actions (CAPA) routing, automated logging/tracking of complaints, automated change control management, or automated procedure management;
- are intended to manage data (process, store, and/or organize data), automate an existing calculation, increase process monitoring, or provide alerts when an exception occurs in an established process; and/or

- are used to support production or the quality system, as explained in Section V.A. above.

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. However, FDA is primarily concerned with the review and assurance for those software features, functions, and operations that are high process risk because a failure also poses a medical device risk. Therefore, for the purposes of this guidance, FDA is presenting the process risks in a binary manner, "high process risk" and "not high process risk." A manufacturer may still determine that a process risk is, for example, "moderate," "intermediate," or even "low" for purposes of determining assurance activities; in such a case, the portions of this guidance concerning "not high process risk" would apply. As discussed in Section V.C. below, assurance activities should be conducted for software that is "high process risk" and "not high process risk" commensurate with the risk.

Example 1: An Enterprise Resource Planning (ERP) Management system contains a feature that automates manufacturing material restocking. This feature ensures that the right materials are ordered and delivered to appropriate production operations. However, a qualified person checks the materials before their use in production. The failure of this feature to perform as intended may result in a mix-up in restocking and delivery, which would be a quality problem because the wrong materials would be restocked and delivered. However, the delivery of the wrong materials to the qualified person should result in the rejection of those materials before use in production; as such, the quality problem should not foreseeably lead to compromised safety. The manufacturer identifies this as an intermediate (not high) process risk and determines assurance activities commensurate with the process risk. The manufacturer already undertakes some of those identified assurance activities so implements only the remaining identified assurance activities.

Example 2: A similar feature in another ERP management system performs the same tasks as in the previous example except that it also automates checking the materials before their use in production. A qualified person does not check the material first. The manufacturer identifies this as a high process risk because the failure of the feature to perform as intended may result in a quality problem that foreseeably compromises safety. As such, the manufacturer will determine assurance activities that are commensurate with the related medical device risk. The manufacturer already undertakes some of those identified assurance activities so implements only the remaining identified assurance activities.

Example 3: An ERP management system contains a feature to automate product delivery. The medical device risk depends upon, among other factors, the correct product being delivered to the device user. A failure of this feature to perform as intended may result in a delivery mix-up, which would be a quality problem that foreseeably compromises safety; as such, the manufacturer identifies this as a high process risk. Since the failure would compromise safety, the manufacturer will next determine the related increase in device risk and identify the assurance activities that are commensurate with the device risk. In this case, the manufacturer

has not already implemented any of the identified assurance activities so implements all of the assurance activities identified in the analysis.

Example 4: An automated graphical user interface (GUI) function in the production software is used for developing test scripts based on user interactions and to automate future testing of modifications to the user interface of a system used in production. A failure of this GUI function to perform as intended may result in implementation disruptions and delay updates to the production system, but in this case, these errors should not foreseeably lead to compromised safety because the GUI function operates in a separate test environment. The manufacturer identifies this as a low (not high) process risk and determines assurance activities that are commensurate with the process risk. The manufacturer already undertakes some of those identified assurance activities so implements only the remaining identified assurance activities.

As noted in FDA's guidance, "30-Day Notices, 135 Day Premarket Approval (PMA) Supplements and 75-Day Humanitarian Device Exemption (HDE) Supplements for Manufacturing Method or Process Changes,"[6] for devices subject to a PMA or HDE, changes to the manufacturing procedure or method of manufacturing that do not affect the safety or effectiveness of the device must be submitted in a periodic report (usually referred to as an annual report).[7] In contrast, modifications to manufacturing procedures or methods of manufacture that affect the safety and effectiveness of the device must be submitted in a 30-day notice.[8] Changes to the manufacturing procedure or method of manufacturing may include changes to software used in production or the quality system. For an addition or change to software used in production or the quality system of devices subject to a PMA or HDE, FDA recommends that manufacturers apply the principles outlined above in determining whether the change may affect the safety or effectiveness of the device. In general, if a change may result in a quality problem that foreseeably compromises safety, then it should be submitted in a 30-day notice. If a change would not result in a quality problem that foreseeably compromises safety, an annual report may be appropriate.

For example, a Manufacturing Execution System (MES) may be used to manage workflow, track progress, record data, and establish alerts or thresholds based on validated parameters, which are part of maintaining the quality system. Failure of such an MES to perform as intended may disrupt operations but not affect the process parameters established to produce a safe and effective device. Changes affecting these MES operations are generally considered annually reportable. In contrast, an MES used to automatically control and adjust established critical production parameters (e.g., temperature, pressure, process time) may be a change to a manufacturing procedure that affects the safety or effectiveness of the device. If so, changes affecting this specific operation would require a 30-day notice.

[6] Available at https://www.fda.gov/regulatory-information/search-fda-guidance-documents/30-day-notices-135-day-premarket-approval-pma-supplements-and-75-day-humanitarian-device-exemption.
[7] 21 CFR 814.39(b), 814.126(b)(1), and https://www.fda.gov/regulatory-information/search-fda-guidance-documents/annual-reports-approved-premarket-approval-applications-pma.
[8] 21 CFR 814.39(b), 814.126(b)(1). Changes in manufacturing/sterilization site or to design or performance specifications do not qualify for a 30-day notice.

C. Determining the Appropriate Assurance Activities

Once the manufacturer has determined whether a software feature, function, or operation poses a high process risk (a quality problem that may foreseeably compromise safety), the manufacturer should identify the assurance activities commensurate with the medical device risk or the process risk. In cases where the quality problem may foreseeably compromise safety (high process risk), the level of assurance should be commensurate with the medical device risk. In cases where the quality problem may not foreseeably compromise safety (not high process risk), the level of assurance rigor should be commensurate with the process risk. In either case, heightened risks of software features, functions, or operations generally entail greater rigor, i.e., a greater amount of objective evidence. Conversely, relatively less risk (i.e., not high process risk) of compromised safety and/or quality generally entails less collection of objective evidence for the computer software assurance effort.

A feature, function, or operation that could lead to severe harm to a patient or user would generally be high device risk. In contrast, a feature, function, or operation that would not foreseeably lead to severe harm would likely not be high device risk. In either case, the risk of the software's failure to perform as intended is commensurate with the resulting medical device risk.

If the manufacturer instead determined that the software feature, function, or operation does not pose a high process risk (i.e., it would not lead to a quality problem that foreseeably compromises safety), the manufacturer should consider the risk relative to the process, i.e., production or the quality system. This is because the failure would not compromise safety, so the failure would not introduce additional medical device risk. For example, a function that collects and records process data for review would pose a lower process risk than a function that determines acceptability of product prior to human review.

Types of assurance activities commonly performed by manufacturers include, but are not limited to, the following:

- **Unscripted testing** – Dynamic testing in which the tester's actions are not prescribed by written instructions in a test case.[9] It includes:
    - **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.[10]
    - **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.[11]

[9] IEC/IEEE/ISO 29119-1 First edition 2013-09-01: *Software and systems engineering – Software testing - Part 1: Concepts and definitions*, Section 4.94.
[10] Ibid., Section 5.6.5.
[11] Ibid., Section 4.14.

- **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 "rules of thumb" regarding common software behaviors and types of failure. Exploratory testing looks for hidden properties, including hidden, unanticipated user behaviors, or accidental use situations that could interfere with other software properties being tested and could pose a risk of software failure.[12]

- **Scripted testing** – Dynamic testing in which the tester's actions are prescribed by written instructions in a test case. Scripted testing includes both robust and limited scripted testing.[13]

  - **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. This approach may apply scripted testing for high-risk features or operations and unscripted testing for low- to medium-risk items as part of the same assurance effort.

In general, FDA recommends that manufacturers apply principles of risk-based testing in which the management, selection, prioritization, and use of testing activities and resources are consciously based on corresponding types and levels of analyzed risk to determine the appropriate activities.[14] 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.

When deciding on the appropriate assurance activities, manufacturers should consider whether there are any additional controls or mechanisms in place throughout the quality system that may decrease the impact of compromised safety and/or quality if failure of the software feature, function or operation were to occur. For example, as part of a comprehensive assurance approach, manufacturers can leverage the following to reduce the effort of additional assurance activities:

- Activities, people, and established processes that provide control in production. Such activities may include procedures to ensure integrity in the data supporting production or software quality assurance processes performed by other organizational units.

[12] Ibid., Section 4.16.
[13] Ibid., Section 4.37.
[14] Ibid., Section 4.35.

- Established purchasing control processes for selecting and monitoring software developers. For example, the manufacturer could incorporate the practices, validation work, and electronic information already performed by developers of the software as the starting point and determine what additional activities may be needed. For some lower-risk software features, functions, and operations, this may be all the assurance that is needed by the manufacturer.

- Additional process controls that have been incorporated throughout production. For example, if a process is fully understood, all critical process parameters are monitored, and/or all outputs of a process undergo verification testing, these controls can serve as additional mechanisms to detect and correct the occurrence of quality problems that may occur if a software feature, function, or operation were to fail to perform as intended. In this example, the presence of these controls can be leveraged to reduce the effort of assurance activities appropriate for the software.

- The data and information periodically or continuously collected by the software for the purposes of monitoring or detecting issues and anomalies in the software after implementation of the software. The capability to monitor and detect performance issues or deviations and system errors may reduce the risk associated with a failure of the software to perform as intended and may be considered when deciding on assurance activities.

- 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.

- The use of testing done in iterative cycles and continuously throughout the lifecycle of the software used in production or as part of the quality system.

For example, supporting software, as referenced in Section V.A., often carries lower risk, such that the assurance effort may generally be reduced accordingly. 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.

Manufacturers are responsible for determining the appropriate assurance activities for ensuring the software features, functions, or operations maintain a validated state. The assurance activities and considerations noted above are some possible ways of providing assurance and are not intended to be prescriptive or exhaustive. Manufacturers may leverage any of the activities or a combination of activities that are most appropriate for risk associated with the intended use.

D. Establishing 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 following:

- 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, including:
    - description of the testing conducted based on the assurance activity;
    - issues found (e.g., deviations, failures) and the disposition;
    - conclusion statement declaring acceptability of the results;
    - the date of testing/assessment and the name of the person who conducted the testing/assessment;
    - established review and approval when appropriate (e.g., when necessary, a signature and date of an individual with signatory authority)

Documentation of assurance activities need not include more evidence than necessary to show that the software feature, function, or operation performs as intended for the risk identified. FDA recommends the record retain sufficient details of the assurance activity to serve as a baseline for improvements or as a reference point if issues occur.[15]

Table 1 provides some examples of ways to implement and develop the record when using the risk-based testing approaches identified in Section V.C. above. Manufacturers may use alternative approaches and provide different documentation so long as their approach satisfies applicable legal documentation requirements.

Table 1 – Examples of Assurance Activities and Records

Assurance Activity	Test Plan	Test Results	Record (Including Digital)
Scripted Testing: Robust	• Test objectives • Test cases (step-by-step procedure) • Expected results • Independent review and approval of test cases	• Pass/fail for test case • Details regarding any failures/deviations found	• Intended use • Risk determination • Detailed report of testing performed • Pass/fail result for each test case • Issues found and disposition • Conclusion statement • Record of who performed testing and date • Established review and approval when appropriate

[15] For the Quality System regulation's general requirements for records, including record retention period, see 21 CFR 820.180.

Assurance Activity	Test Plan	Test Results	Record (Including Digital)
Scripted Testing: Limited	• Limited test cases (step-by-step procedure) identified • Expected results for the test cases • Identify unscripted testing applied • Independent review and approval of test plan	• Pass/fail for test case identified • Details regarding any failures/deviations found	• Intended use • Risk determination • Summary description of testing performed • Pass/fail test result for each test case • Issues found and disposition • Conclusion statement • Record of who performed testing and date • Established review and approval when appropriate
Unscripted Testing: Ad-hoc	• Testing of features and functions with no test plan	• Details regarding any failures/deviations found	• Intended use • Risk determination • Summary description of features and functions 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
Unscripted Testing: Error guessing	• Testing of failure-modes with no test plan	• Details regarding any failures/ deviations found	• 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
Unscripted Testing: Exploratory Testing	• Establish high level test plan objectives (no step-by-step procedure is necessary)	• Pass/fail for each test plan objective • Details regarding any failures/deviations found	• Intended use • Risk determination • Summary description of the objectives tested and testing performed • Pass/fail test result for each objective • Issues found and disposition • Conclusion statement • Record of who performed testing and date of testing • Established review and approval when appropriate

The following is an example of a record of assurance in a scenario where a manufacturer has developed a spreadsheet with the intended use of collecting and graphing nonconformance data stored in a controlled system for monitoring purposes. In this example, the manufacturer has established additional process controls and inspections that ensure non-conforming product is not released. In this case, failure of the spreadsheet to perform as intended would not result in a quality problem that foreseeably leads to compromised safety, so the spreadsheet would not pose a high process risk. The manufacturer conducted rapid exploratory testing of specific functions used in the spreadsheet to ensure that analyses can be created, read, updated, and/or deleted. During exploratory testing, all calculated fields updated correctly except for one deviation that occurred during update testing. In this scenario, the record would be documented as follows:

- **Intended Use:** The spreadsheet is intended for use in collecting and graphing nonconformance data stored in a controlled system for monitoring purposes; as such, it is used as part of production or the quality system. Because of this use, the spreadsheet is different from similar software used for business operations such as for accounting.

- **Risk-Based Analysis:** In this case, the software is only used to collect and display data for monitoring nonconformances, and the manufacturer has established additional process controls and inspections to ensure that nonconforming product is not released. Therefore, failure of the spreadsheet to perform as intended should not result in a quality problem that foreseeably leads to compromised safety. As such, the software does not pose a high process risk, and the assurance activities should be commensurate with the process risk.

- **Tested:** Spreadsheet X, Version 1.2

- **Test type:** Unscripted testing – exploratory testing

- **Goal:** Ensure that analyses can be correctly created, read, updated, and deleted

- **Testing objectives and activities:**

    - Create new analysis – Passed
    - Read data from the required source – Passed
    - Update data in the analysis – Failed due to input error, then passed
    - Delete data – Passed
    - Verify through observation that all calculated fields correctly update with changes – Passed with noted deviation

- **Deviation:** During update testing, when the user inadvertently input text into an updatable field requiring numeric data, the associated row showed an immediate error.

- **Conclusion:** No errors were observed in the spreadsheet functions beyond the deviation. Incorrectly inputting text into the field is immediately visible and does not impact the risk of the intended use. In addition, a validation rule was placed on the field to permit only numeric data inputs.

- **When/Who:** July 9, 2019, by Jane Smith

Advances in digital technology may allow for manufacturers to leverage automated traceability, testing, and the electronic capture of work performed to document the results, reducing the need for manual or paper-based documentation. 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, as opposed to paper documentation and screenshots, in establishing the record associated with the assurance activities.

Manufacturers have expressed confusion and concern regarding the application of Part 11, Electronic Records; Electronic Signatures, to computers or automated data processing systems used as part of production or the quality system. As described in the "Part 11, Electronic Records; Electronic Signatures – Scope and Application" guidance,[16] the Agency intends to exercise enforcement discretion regarding Part 11 requirements for validation of computerized systems used to create, modify, maintain, or transmit electronic records (see 21 CFR 11.10(a) and 11.30). In general, Part 11 applies to records in electronic form that are created, modified, maintained, archived, retrieved, or transmitted under any records requirements set forth in Agency regulations (see 21 CFR 11.1(b)). Part 11 also applies to electronic records submitted to the Agency under requirements of the Federal Food, Drug, and Cosmetic Act (FD&C Act) and the Public Health Service Act (PHS Act), even if such records are not specifically identified in Agency regulations (see 21 CFR 11.1(b)).

In the context of computer or automated data processing systems, for computer software used as part of production or the quality system, a document required under Part 820 and maintained in electronic form would generally be an "electronic record" within the meaning of Part 11 (see 21 CFR 11.3(b)(6)). For example, if a document requires a signature under Part 820 and is maintained in electronic form, then Part 11 applies (see, e.g., 21 CFR 820.40 (requiring signatures for control of required documents)).

[16] https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application.

Appendix A. Examples

The examples in this section outline possible application of the principles in this draft guidance to various software assurance situations cases.

Example 1: Nonconformance Management System

A manufacturer has purchased COTS software for automating their nonconformance process and is applying a risk-based approach for computer software assurance in its implementation. The software is intended to manage the nonconformance process electronically. The following features, functions, or operations were considered by the manufacturer in developing a risk-based assurance strategy:

Table 2. Computer Software Assurance Example for a Nonconformance Management System

Features, Functions, or Operations	Intended Use of the Features, Functions or Operations	Risk-Based Analysis	Assurance Activities	Establishing the appropriate record
Nonconformance (NC) Initiation Operations: • A nonconforming event results in the creation of an NC record. • The necessary data for initiation are recorded prior to completion of an NC initiation task. • An NC Owner is assigned prior to completion of the NC initiation task.	The intended uses of the operations are to manage the workflow of the nonconformance and to error-proof the workflow to facilitate the work and a complete quality record. These operations are intended to supplement processes established by the manufacturer for containment of non-conforming product.	Failure of the NC initiation operation to perform as intended may delay the initiation workflow, but would not result in a quality problem that foreseeably compromises safety, as the manufacturer has additional processes in place for containment of non-conforming product. As such, the manufacturer determined the NC initiation operations did not pose a high process risk.	The manufacturer has performed an assessment of the system capability, supplier evaluation, and installation activities. In addition, the manufacturer supplements these activities with exploratory testing of the operations. High level objectives for testing are established to meet the intended use and no unanticipated failures occur.	The manufacturer documents: • the intended use • risk determination, • summary description of the features, functions, operations tested • the testing objectives and if they passed or failed • any issues found and their disposition • a concluding statement noting that the performance of the operation is acceptable • the date testing was performed, and who performed the testing.

Features, Functions, or Operations	Intended Use of the Features, Functions or Operations	Risk-Based Analysis	Assurance Activities	Establishing the appropriate record
Electronic Signature Function: • The electronic signature execution record is stored as part of the audit trail. • The electronic signature employs two distinct identification components of a login and password. • When an electronic signature is executed, the following information is part of the execution record: o The name of the person who signs the record o The date (DD-MM-YYYY) and time (hh:mm) the signature was executed. o The meaning associated with the signature (such as review, approval, responsibility, or authorship).	The intended use of the electronic signature function is to capture and store an electronic signature where a signature is required and such that it meets requirements for electronic signatures.	If the electronic signature function were to fail to perform as intended, then production or quality system records may not reflect appropriate approval or be sufficiently auditable, or may fail to meet other regulatory requirements. However, such a failure would not foreseeably lead to compromised safety. As such, the manufacturer determined that this function does not pose high process risk.	The manufacturer has performed an assessment of the system capability, supplier evaluation, and installation activities. To provide assurance that the function complies with applicable requirements, the manufacturer performs ad-hoc testing of this function with users to demonstrate the function meets the intended use.	The manufacturer documents: • the intended use • risk determination • testing performed • any issues found and their disposition • a concluding statement noting that the performance of the function is acceptable • the date testing was performed and who performed the testing.

Features, Functions, or Operations	Intended Use of the Features, Functions or Operations	Risk-Based Analysis	Assurance Activities	Establishing the appropriate record
Product Containment Function: • When a nonconformance is initiated for product outside of the manufacturer's control, then the system prompts the user to identify if a product correction or removal is needed.	This function is intended to trigger the necessary evaluation and decision-making on whether a product correction or removal is needed when the nonconformance occurred in product that has been distributed.	Failure of the function to perform as intended would result in a necessary correction or removal not being initiated, resulting in a quality problem that foreseeably compromises safety. The manufacturer therefore determined that this function poses high process risk.	The manufacturer has performed an assessment of the system capability, supplier evaluation, and installation activities. Since the manufacturer determined the function to pose high process risk, the manufacturer determined assurance activities commensurate with the medical device risk: established a detailed scripted test protocol that exercises the possible interactions and potential ways the function could fail. The testing also included appropriate repeatability testing in various scenarios to provide assurance that the function works reliably.	The manufacturer documents: • the intended use • risk determination • detailed test protocol developed • detailed report of the testing performed • pass/fail results for each test case • any issues found and their disposition • a concluding statement noting that the performance of the operation is acceptable • the date testing was performed and who performed the testing • the signature and date of the appropriate signatory authority.

Example 2: Learning Management System (LMS)

A manufacturer is implementing a COTS LMS and is applying a risk-based approach for computer software assurance in its implementation. The software is intended to manage, record, track, and report on training. The following features, functions, or operations were considered by the manufacturer in developing a risk-based assurance strategy:

Table 3. Computer Software Assurance Example for an LMS

Features, Functions, or Operations	Intended Use of the Features, Functions or Operations	Risk-Based Analysis	Assurance Activities	Establishing the appropriate record
• The system provides user log-on features (e.g., username and password) • The system assigns trainings to users per the curriculum assigned by management • The system captures evidence of users' training completion • The system notifies users of training curriculum assignments, completion of trainings, and outstanding trainings • The system notifies users' management of outstanding trainings • The system generates reports on training curriculum assignments, completion of training, and outstanding trainings	All of the features, functions, and operations have the same intended use, that is, to manage, record, track and report on training. They are intended to automate processes to comply with 21 CFR 820.25 (Personnel), and to establish the necessary records.	Failure of these features, functions, or operations to perform as intended would impact the integrity of the quality system record but would not foreseeably compromise safety. As such, the manufacturer determined that the features, functions, and operations do not pose high process risk.	The manufacturer has performed an assessment of the system capability, supplier evaluation, and installation activities. In addition, the manufacturer supplements these activities with unscripted testing, applying error-guessing to attempt to circumvent process flow and "break" the system (e.g. try to delete the audit trail).	The manufacturer documents: • the intended use • risk determination • a summary description of the failure modes tested • any issues found and their disposition • a concluding statement noting that the performance of the operation is acceptable • the date testing was performed, and who performed the testing.

Example 3: Business Intelligence Applications

A medical device manufacturer has decided to implement a commercial business intelligence solution for data mining, trending, and reporting. The software is intended to better understand product and process performance over time, in order to provide identification of improvement opportunities. The following features, functions, or operations were considered by the manufacturer in developing a risk-based assurance strategy:

Table 4. Computer Software Assurance Example for a Business Intelligence Application

Features, Functions, or Operations	Intended Use of the Features, Functions or Operations	Risk-Based Analysis	Assurance Activities	Establishing the appropriate record
Connectivity Functions: • The software allows for connecting to various databases in the organization and external data sources. • The software maintains the integrity of the data from the original sources and is able to determine if there is an issue with the integrity of the data, corruption, or problems in data transfer.	These functions are intended to ensure a secure and robust capability for the system to connect to the appropriate data sources, ensure integrity of the data, prevent data corruption, modify, and store the data appropriately.	Failure of these functions to perform as intended would result in inaccurate or inconsistent trending or analysis. This would result in failure to identify potential quality trends, issues or opportunities for improvement, which in some cases, may result in a quality problem that foreseeably compromises safety. As such, the manufacturer determined that these functions posed high process risk, necessitating more-rigorous assurance activities, commensurate with the related medical device risk.	The manufacturer determined assurance activities commensurate with the medical device risk and has performed an assessment of the system capability, supplier evaluation, and installation activities. Additionally, the manufacturer establishes a detailed scripted test protocol that exercises the possible interactions and potential ways the functions could fail. The testing also includes appropriate repeatability testing in various scenarios to provide assurance that the functions work reliably.	The manufacturer documents: • the intended use • risk determination • detailed test protocol • a detailed report of the testing performed • pass/fail results for each test case • any issues found and their disposition • a concluding statement noting that the performance of the operation is acceptable • the date testing was performed, and who performed the testing • the signature and date of the appropriate signatory authority.

Features, Functions, or Operations	Intended Use of the Features, Functions or Operations	Risk-Based Analysis	Assurance Activities	Establishing the appropriate record
Usability Feature: • The software provides the user a help menu for the application.	This feature is intended to facilitate the interaction of the user with the system and provide assistance on use of all the system features.	The failure of the feature to perform as intended is unlikely to result in a quality problem that would lead to compromised safety. Therefore, the manufacturer determined that the feature does not pose high process risk.	The feature does not necessitate any additional assurance effort beyond what the manufacturer has already performed in assessing the system capability, supplier evaluation, and installation activities.	The manufacturer documents: • the intended use • risk determination • the date of assessment and who performed the assessment • a concluding statement noting that the performance is acceptable given the intended use and risk.
Reporting Functions: • The software is able to create and perform queries and join data from various sources to perform data mining. • The software allows for various statistical analysis and data summarization. • The software is able to create graphs from the data. • The software provides the capability to generate reports of the analysis.	These functions are intended to allow the user to query the data sources, join data from various sources, perform analysis, and generate visuals and summaries. These functions are intended for collection and recording data for monitoring and review purposes that do not have a direct impact on production or process performance. In this example, the software is not intended to inform quality decisions.	Failure of these functions to perform as intended may result in a quality problem (e.g., incomplete or inadequate reports) but, in this example, would not foreseeably lead to compromised safety because these functions are intended for collection and recording data for monitoring and review purposes that do not have a direct impact on production or process performance. Therefore, the manufacturer determined that these functions do not pose high process risk.	The supplier of the reporting software has validated the ability of the software to create and perform queries, join data from various sources to perform data mining, perform statistical analysis and data summarization, create graphs and generate reports. Beyond this, the manufacturer has assessed the system capability and performed supplier evaluation and installation activities. As such, the manufacturer determined that the reporting functions of the software do not necessitate any additional assurance effort beyond these activities.	The manufacturer documents: • the intended use • risk determination • the date of assessment and who performed the assessment • a concluding statement noting that the performance is acceptable given the intended use and risk.

632