

Procedure For Software Quality Assurance

1 Introduction

This is the metrology laboratory procedure for protecting, validating, and approving the accuracy of computer software and systems. All software and systems that affect reported measurement results, reported corrections, or uncertainties must be evaluated to comply with this document. All supporting software used in the laboratory to monitor the validity of measurement results must be validated as well.

The verification and validation process must occur at all phases of software life cycle (Figure 1). In addition to validating and verifying software, it is important to assess the knowledge, skills, and attitude of the metrologist and staff to ensure proper use and application of the software to ensure that no inadvertent measurement errors are introduced due to poor data entry, improperly validated software modifications, or general use in the laboratory.

The use/configuration of Commercial Off-The-Shelf (COTS) software in a laboratory is, by definition, considered software engineering, and must comply with good software engineering practices including these verification and validation methods. (Note: this includes but is not limited to spreadsheets like Excel^{TM1} and coding in programs like Visual Basic.)

2 Purpose

The purpose of this procedure is to ensure that software and systems do not contribute errors or additional uncertainty to any measurement process and to ensure that computer systems are adequately designed, developed, and secure; this is called Software Verification. Additionally, software is evaluated to make sure that the software is designed to comply with the requirements of the chosen Standard Operating Procedure (SOP) for calibration, uncertainty, and reporting; this is called Software Validation. Use of this procedure is designed to increase the usability and reliability of software used in the laboratory, increase the quality of work done, and reduce liability.

3 Responsibility and authority

3.1 For COTS software that does not have built in protections to prevent accidental changes during routine use, the Technical Manager implements a system of templates, cell protection,

¹ No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial equipment, instruments, or materials are identified in this paper to facilitate understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

read only access, or other security measures to protect the local configuration.

The Quality Manager/Laboratory Director is responsible for ensuring that this Standard Administrative Procedure (SAP) is followed and documented, and that all software associated with a measurement result has been validated. The Quality Manager/Laboratory Director is responsible to validate, or arrange to have another metrologist validate, all software. The Technical Manager ensures that numerical computations are correct, systems and software are adequately documented through instructions and/or manuals (at the level needed for accurate staff use), and presents evidence to the Laboratory Supervisor for approval using Form A (file associated with this procedure). It is important to note that self-validation is extremely difficult, thus in smaller laboratories, additional steps need to be taken to ensure good verification and validation of software. Additional steps may include interlaboratory evaluations of procedural files.

3.2 The Laboratory Director or Technical Manager coordinates with Information Technology staff (IT) regarding all laboratory software and computer requirements for the laboratory, including but not limited to updates to operating systems, operating software, network access, backup policies, file storage, file access, and retention.

3.3 The Technical Manager employs the following security measures to maintain the security of the disks where primary software resides to ensure that systems and software are protected from unauthorized access, safeguarded against tampering and loss; operated in an environment that complies with provider or laboratory specifications, maintained in such a manner as to ensure the integrity of data and information, and methods for ensuring data is not lost through system failures:

3.3.1 Metrology laboratory staff and Laboratory Director who have been trained in specific applications and procedures who should have access to the applicable disks/files are specified.

3.3.2 The IT staff will have access to files for emergency purposes, but are not authorized to perform software changes nor validate technical applications without suitable metrology training or technical knowledge of the procedure.

3.3.3 Regular backups are performed weekly (all files) and daily (modified files).

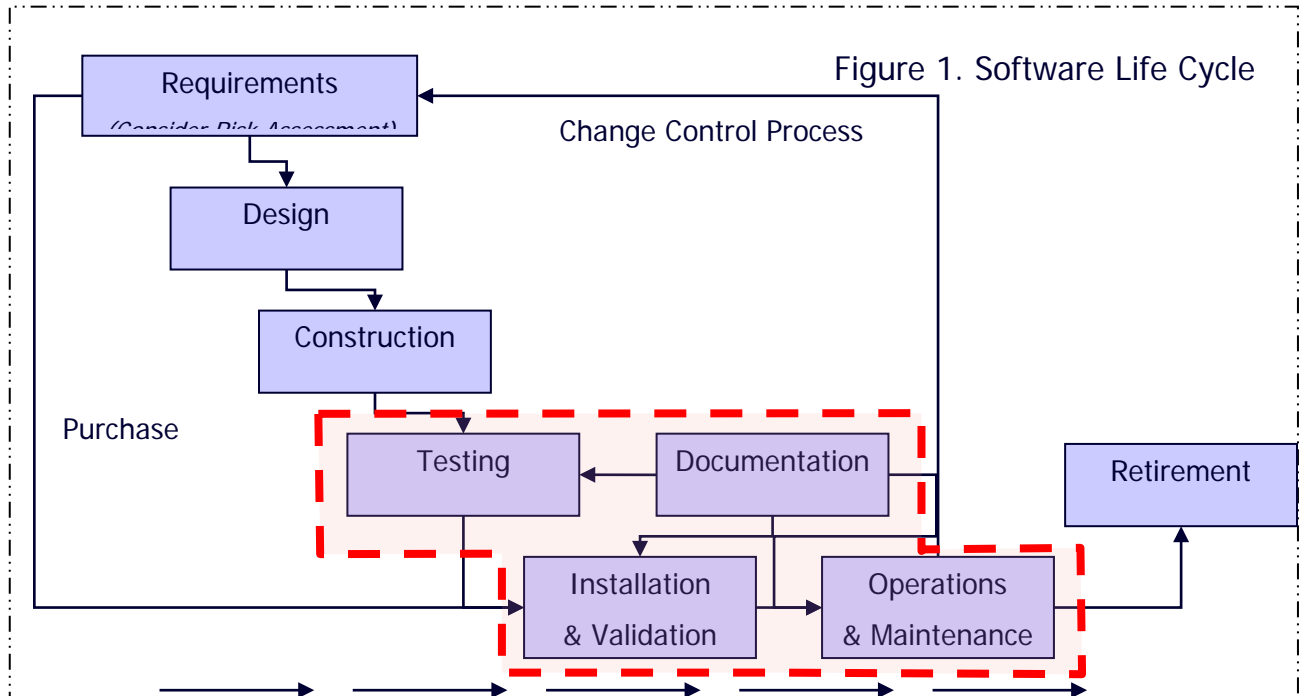
3.3.4 Restoration of software in the event of a disk failure.

3.4 Commercial Off-The-Shelf software packages are commonly used by laboratories and are sufficiently validated for all use; however, each laboratory must choose the software appropriately. Calculation results from commercial software must be evaluated during software testing and approval. Some spreadsheet functions, such as rounding or even simple calculations with large numbers and small differences, may not be adequate for the reporting of calibration results and must comply with the accuracy requirements of the procedure as well as other good laboratory practices. The modification and use of COTS spreadsheets in the laboratory are considered software engineering and must be validated.

A list of COTS software is maintained in the laboratory's Inventory and Assessment file (see Form B a file associated with this procedure).

4 Software Engineering

Good software engineering includes phases for setting requirements, designing, constructing, testing, installing, validating, documenting, performing operations and maintenance, and retiring the software when appropriate. All phases are important aspects to consider when validating software. See the software life cycle (Figure 1) for the workflow. ²Form A methods and questions should be considered during all phases of the software life cycle.



4.1 Software Life Cycle

4.1.1 Requirements Phase

This phase identifies, specifies, analyzes, and documents all the requirements that the software must satisfy regarding functionality, performance, design constraints, attributes, and external interfaces. It is important to perform a risk assessment during this phase.

4.1.2 Design Phase

This phase develops, documents, and reviews a design that satisfies the requirements previously documented.

4.1.3 Construction Phase

This phase takes each element documented in the design phase and translates it into a programming language, and may incorporate COTS software or rely on it solely. This phase is often known as a “coding” or “build” phase.

² NCSLI Recommended Practice (RP) 13, 1996. Figure 1 as modified.

4.1.4 Testing Phase

This phase runs the software through test cases and analyzes any failure to determine which phase contributed to such error.

4.1.5 Installation and Validation Phase

This phase executes tests for the installation and integration of the software into the equipment (i.e., other software, data, hardware), and the documentation of the approval of the software for operational use. User site testing is a very important practice to consider because it helps eliminate errors (i.e., bugs) that may arise after the software is installed on different equipment. Form A (file associated with this procedure) must be used to ensure complete assessment of all aspects of the software (especially ensuring that calculation functions are not the only item assessed) and to begin the documentation.

4.1.6 Documentation Phase

This phase handles the technical documentation of all the phases described above. Evidence must be retained in association with the methods used in Form A (file associated with this procedure).

4.1.7 Operations and Maintenance

Once the software has been approved for operational use, routine maintenance may be performed to remove errors, to respond to new or modified equipment, or to adapt the software to changes in the operating environment. All planned changes must be approved by the Laboratory Director or Technical Manager before work is started. After any modifications, software must be verified and validated again.

5 Risk Analysis

Software is considered of higher risk when calibration measurement results and uncertainty values are used on calibration certificates and provided to the customer.

Where the software is developed by someone else, for example, configuration of COTS software, the final user (laboratory) must study the information provided by the supplier to properly assess the risk of usage, as the supplier may have a different application in mind, errors may not have been discovered, or software may not be completely validated.³

6 Methods and Practices for Evaluation

6.1 Documentation

Documentation of the verification and validation of all computer systems and software is maintained in the laboratory. These files include an inventory of laboratory computers, COTS software, laboratory-developed software, and a copy of the “Inventory and Assessment” Excel file.

³ For additional information on Risk Analysis practices and procedures, see Validation of software in measurement systems (Software for Metrology Best Practice Guide No. 1), National Physical Laboratory (NPL), <http://www.npl.co.uk/>.

Each laboratory spreadsheet has a worksheet/tab for document control/revisions, a worksheet/tab for instructions, and verification/validation worksheet/tab (or the verification/validation is maintained in other appropriate files). Revisions are documented in each file and include: version control, details of changes that have been made and indications of staff approvals. Instructions include how to use the software and reference any prerequisite training or knowledge staff must have to use the software. The verification/validation tab may include the Technical Assessment of Software (Form A) or separate documentation may be retained in the laboratory.

6.2 Assessment

Technical assessment of software includes, but may not be limited to, completion of Form A (file associated with this procedure), with all associated objective evidence retained with the summary evaluations.

This assessment is to be performed during the design and development phase, upon installation, whenever operating systems change, when software is moved, and when software is updated and must be verified at all work stations.

6.3 Assessment Methods and Examples of Evidence

The following table provides a brief overview of the assessment methods used to evaluate software and complete Form A.

Codes for Validation		Descriptions	Evidence
A	Software inspection	Review the software. Is it clear and does it make sense? Are there instructions for use? Are data entry fields labeled and color coded? Is it obvious what procedure is being used? Is there adequate documentation for a metrologist who is trained in this procedure to know what and where data is entered? Are they able to ensure that no data entry is inadvertently left out? Is the spreadsheet “blank” when opened to make sure old data is not accidentally used? Is there traceability to the specifications document (often an SOP)? Are cells formatted appropriately? Are unused cells locked? Are unused sheets removed? Are worksheets named appropriately? (Basically, were good spreadsheet design concepts followed?)	Evidence: describe the review that was conducted in a few sentences.
B	Mathematical specification	Is the correct SOP used? Are the correct formulae selected? E.g., SOP 2 for air density has 2 formulas and one is recommended – which one was used? Is it the one you want used at the reported level of precision/uncertainty? Is there direct traceability of the equations?	Evidence: Include a “documentation” worksheet in your workbooks that identify which SOP and equations are used.
C	Code review	Compare the cells with the formulae line by line in the spreadsheet versus the SOP. Do they match exactly? Are repeated calculations copied	Evidence: Save one of the worksheets in your

Codes for Validation		Descriptions	Evidence
		exactly or appropriately referencing the correct cells? Is rounding done at the appropriate locations in the file?	workbook with the equations showing and notes included to show that they were evaluated – include a “reference” column to show which section of the SOP was compared. E.g., “compared to equation 3.2.2. in SOP 4.” Include a graphic capture of the equation to support the comparison further.
D	Numerical stability	This component of Excel needs to be evaluated for rounding practices and the stability of precision calculations. See the NPL example in Best Practice Guide Number 1.	
E	Component testing	Components include things like Pass/Fail tests, color coding, automatic look-ups for standards or uncertainties from a master list/table or master file. They might include automatic report generation macros. Depending on the component, you will need to create different kinds of approaches for the components and the evidence may include a description or saved examples. Have you tested the functionality of each functional macro? Each command/button? Combinations of interdependent macros? Accuracy of plotted graphs? Printing of each printable worksheet/report?	Evidence: Pass/Fail: intentionally enter good or bad data to see if the criteria changes. Conditional color formatting: most often used as a pass/fail or marginal flag – enter good, bad, and marginal data to determine the response. Look up tables: sample (or do 100% evaluations) the look up tables to make sure items are selected from the right row, column, cell, and workbook. Make sure data is transferred accurately for automation from laboratory instruments and transferred accurately to the final calibration report.
F	Numerical reference results	Two aspects need to be considered here: 1. All data in tables and lists match their reference sources exactly; and 2. All values with automatic look up features need to select the correct row, column, cell, and workbook. Do look up tables and lists match the latest calibration report? Do uncertainties match the latest Scope? If values reference another workbook or spreadsheet, is it dated – and if you update the date on a master list of standards, does the file reference a default value, old value, or zero instead of an error message?	Evidence: Print reference values and do line by line comparisons to ensure they match. Alternatively, scan a graphic and do the line by line comparison in the workbook.

Codes for Validation		Descriptions	Evidence
G	Embedded data evaluation	<p>Sometimes conversion factors, reference values, or other mathematical factors are included in a calculation. E.g., air density and water density equations have many standard multipliers with many decimal places.</p> <p>Conversion factors need to be the most accurate ones available and need to be rounded, when appropriate, to the right number of digits to avoid impacting the final results. Even without look-up values, some values for standards or uncertainties may be embedded in a working file and the accuracy of those values must be ensured.</p>	<p>Evidence:</p> <p>Identify all embedded values in a list (ideally within the spreadsheet) and compare them to the correct reference values and note the date of the comparison. The dates will help ensure that if subsequent conversion factors are used or standards are calibration the right values are entered.</p>
H	Back-to-back testing	<p>Data that is published in an SOP, generated by the laboratory, used for a proficiency test, or even simply created for testing purposes may be used.</p> <p>Do two spreadsheets – created by different people – perhaps in different software – agree? Does a newer spreadsheet agree with an older spreadsheet down to the level of intermediate calculations? If there are differences, do they agree well beyond the level where they could impact the uncertainty of the calibration?</p>	<p>Evidence:</p> <p>Be sure to save both spreadsheet files and note file names and dates of evaluation.</p>
I	Analysis without computer assistance (data sets)	<p>Data that is published in an SOP, generated by the laboratory, used for a proficiency test, or even simply created for testing purposes may be used.</p> <p>Do hand calculations with a scientific calculator (hand or “computer calculator”) agree with those generated by the spreadsheet?</p>	<p>Evidence:</p> <p>Be sure to save the spreadsheet file and a copy of the hand-written notes; record the dates of evaluation.</p>
J	Security	<p>Can a metrologist accidentally delete equations and calculation cells that should be protected? Can cells be accidentally moved around? Is it possible for an untrained metrologist to “correct” something by mistake because passwords are readily available?</p> <p>How are all files backed up? Is there a source of the back-up files maintained in an alternate facility/location? Can the files on network drives accidentally be deleted? If a computer fails (or facility damaged where the computer can no longer be used, is there a back-up somewhere?</p>	<p>Evidence:</p> <p>Describe the review that was conducted in a few sentences.</p>

In addition to following this procedure in the laboratory, the software development process assumes the knowledge and familiarity of the operator with this procedure and with the applicable procedure

being evaluated. It is critical for the operator to pay attention while using any software to find potential data-entry errors as they occur. One might call this real-time validation based on the operator's experience, knowledge, and judgment.

The level of confidence, therefore the level of software validation, verification, and testing effort needed, varies depending upon the risk posed by the software.