Software validation applied to spreadsheets used in laboratories working under ISO/IEC 17025

J M Banegas [1,2], **M W Orué** [1,2]

[1] Laboratorio de Metrología del Litoral (LabMET) – UTN Santa Fe / CCT Santa Fe; [2] Grupo Control y Seguridad Eléctrica (CYSE) – UTN Santa Fe

E-mail: jmbanegas@frsf.utn.edu.ar

Abstract: Several documents deal with software validation. Nevertheless, more are too complex to be applied to validate spreadsheets; surely the most used software in laboratories working under ISO/IEC 17025. The method proposed in this work is intended to be directly applied to validate spreadsheets. It includes a systematic way to document requirements, operational aspects regarding to validation, and a simple method to keep records of validation results and modifications history. This method is actually being used in an accredited calibration laboratory, showing to be practical and efficient.

Keywords: software validation, spreadsheet calculations, ISO/IEC 17025.

1. INTRODUCTION

Different kind of software is widely used in ISO/IEC 17025 accredited laboratories to perform a variety of tasks. Some of them are as simple and not critical as writing a document. In other cases, the use of software involves critical data processing and registering raw data; in such cases, some aspects like confidence, access security, and risk assessment, start to matter.

In order to cover this broad range of activities, laboratories use many different software, ranging from standard software packages, to custom made programs. Within the whole universe of available software, spreadsheets are of particular interest, since virtually every laboratory – metrology labs in special – uses them to process data, and calculate uncertainties.

The core requirements about the use of computers are in ISO/IEC 17025:2005 [1], clause 5.4.7.2. In particular, it states that “the laboratory shall ensure that… computer software developed by the user is documented in sufficient detail and suitably validated as being adequate for use”. Regarding commercial off the shelf software (COTS) – including spreadsheets – the Note clarifies that the software itself “in general use within its designed application range may be considered sufficiently validated. However, laboratory software configuration/modifications should be validated…”.

Validation is defined in [1] clause 5.4.5.1, as the confirmation by examination and the provision of objective evidence that requirements for a specific application are fulfilled [1]. Similar definitions are given in others standards like ISO 9000 [2] and the VIM [3]. Thus, validation implies three main aspects: documented software requirements for the specific application, evaluation process (namely, validation procedure), and records to provide evidence.

Although a recognized spreadsheet doesn’t need validation itself, the formulas, logic, programming, etc., must be validated, since they are configurations/modifications performed by the laboratory. Even more, aspects like conditional formatting and cell protection should be taken into account, if they are necessary to guarantee the adequacy for the intended use of the spreadsheet. Usually, this is called a *modified off the shelf* software (MOTS) [i.e. 4, 5].

Many documents provide guidance on software validation and verification, but only a few are intended to be applicable to laboratories working under ISO/IEC 17025 [i.e. 4-7].

The aim of this work is to present a straight-forward validation method specifically applicable to spreadsheets used in laboratories, including software specifications documentation, validation itself, and associated records. This method is being successfully applied in an accredited laboratory, showing to be practical and simple.

2. SOFTWARE LIFE CYCLE AND VALIDATION

Software life cycle is closely related to validation. The approach presented in the standard IEEE 1012 considers validation and verification processes “performed in parallel with all life cycle stages, not at their conclusion” [8]. This approach is shared in other publications [9].

On the other hand, validation in ISO/IEC 12207 is one of the “software specific processes”, and it’s mostly performed at the end of the development [10].

In any case, it’s clear that a suitable model of the software life cycle has to be adopted. Most of the models depicted in the references are too complex to be used with a spreadsheet. Nevertheless, three typical stages can be clearly identified: development, operation and maintenance. This is probably the simplest software life cycle model, and it’s the model adopted in this work.

First, validation takes part in the development stage. The earlier part of development is to establish the requisites and objectives for the spreadsheet, taking into account its intended use. It also provides the basis for the validation, namely, the software requirements to be evaluated. And finally, regarding of other verifications performed during the spreadsheet programming, the development stage must finish with the validation, before operation.

Later, in the maintenance stage, full or partial validation must be performed every time a new version of the spreadsheet is to be released.

3. VALIDATION ASPECTS AND PROCEDURE

Usually a particular spreadsheet is modified and upgraded several times during its life cycle. Keeping track of the changes and the validation records of each version can be a hard task. Moreover, as the calibration procedures are updated, the requirements for the associated spreadsheets may change.

In order to simplify the process and minimize the paperwork, the proposed method uses the spreadsheet file itself for documentation. It is done by adding to the workbook a dedicated worksheet for validation. This worksheet is used for requirements documentation, validation records, and history tracking, as stated below.

A typical ‘validation’ worksheet would look as shown in figure 1.

ID	Requisite	Cply.	Notes

Figure 1. Example of ‘validation’ worksheet.

3.1. Software requirements

Spreadsheet requirements regarding behavior and expected results must be clearly established in order to perform validation.

Many requirements are derived from the applicable documented calibration procedures.

These describe how data has to be processed (including formulas, coefficients, constants, etc.) and how the results are presented. Other requirements may be specific for each spreadsheet. In any case, they all must be documented. Requisites should be grouped in categories as follows:

- Behaviour (B): functional aspects of the spreadsheet, i.e. conditional formatting, date/time formatting, proper functioning of lists for selectable data (combos), etc.
- Formulas and calculations (F): calculations the spreadsheet is intended to perform, including factors, coefficients, etc.
- Logic and automation (L): automated actions the spreadsheet is intended to perform (i.e. allow/block data entry in specific cells, search in a matrix, select data), when particular conditions are fulfilled (i.e. data selected in a combo, data entered with certain values).
- Security and protection (S): access level to sheets and/or cells, in order to avoid undue modifications to the spreadsheet.

Requisites should be written in the 'validation' worksheet as short and as explicit as possible. If necessary, they may have references to the respective documented procedure. Each individual requisite should be coded (category and sequential number) in order to facilitate referencing. Here are some examples:

B01. *Combo 'Working standard' lists every possible instrument and works OK.*

B02. *'Emission date' cell painted in red if previous than 'Cal date'.*

F01. *Error formula in cells ZBxx according to procedure formula (3).*

F02. *Results as expected, with typical data set.*

L01. *Correct selection of set of specifications according sensor type.*

L02. *Multiplies reference value and uncertainties x50 when 'coil' used.*

S01. *Every worksheet protected.*

S02. *Cells for data entry left unprotected.*

3.2. Examination procedure

Every single requirement must be evaluated to compliance, recording the results in the 'compliance' column of the 'validation' worksheet. Depending on the requisite, a black-box or a white-box testing [11] is used.

In a **black-box** (or functional) test, results are evaluated in response to inputs, ignoring the internal working of the software. It's very useful to evaluate compliance with specified functional requirements. For instance, requisites B01, B02, F02, L01, S01, and S02, are checked by black-box testing.

The procedure is checking the results for consistency and compliance, when individual actions – like selecting data from combos, introducing known set of values, etc. – are performed. It is important to notice that embedded spreadsheet formulas (i.e. LOG, SUM, STDEV) are considered sufficiently validated, thus no further validation is required.

The first validation of a particular spreadsheet should always include a black-box validation consisting of introducing a typical data set with known output results. Data should include boundary points, and even unexpected values.

In a **white-box** (or structural) test, internal working of the software is taken into account. In spreadsheets this can be done evaluating the algorithms and formulas used, checking for adequacy. For instance, requisites F01 and L02 should be checked by white-box testing, namely, analyzing if the formulas are written correctly and refer to the right cells.

Both types of testing are complementary, and sometimes one particular requirement may be evaluated by any of them. The laboratory must establish the method of evaluation for each requirement.

3.3. Evaluation records

The spreadsheet itself constitutes the evidence of the validation, since the ‘validation’ worksheet contains both specifications and results.

In case of a full validation – or in any case that the spreadsheet was loaded with data to perform black-box testing – it’s necessary to save the filled spreadsheet as evidence. A blank copy is then used for everyday work.

3.4. Re-validation triggers

Every time the spreadsheet is modified, the changes have to be validated before use. If changes are small, only a partial validation can be done, covering the requirements that could have been affected by the modifications.

The laboratory must keep the versioning history, specifying changes between versions, and thus establishing the requisites to re-validate. It can be easily done in the same ‘validation’ worksheet, or using a dedicated one. History must include the last full-validation date, and the validations performed in each new version since then (i.e. the requisites that were re-evaluated).

5. CONCLUSIONS

Spreadsheets used for processing raw calibration or testing data must be validated, before use and after any modification.

The proposed method covers the 3 main aspects of validation: requirements, examination and records. Paperwork is kept to a minimum, since documentation of requirements and validation records are part of the spreadsheet itself, and no additional documents are required.

It also shows a simple way to re-validate modified spreadsheets, keeping track of history and performing only partial validations. Thus, the method is flexible and efficient.

This method is being successfully used in an accredited metrology lab. Several spreadsheets – covering calibrations in the field of temperature, time, electricity, pressure, and humidity – were validated and re-validated, showing the method is practical, simple and affective.

6. REFERENCES

[1] ISO/IEC 17025:2005, *General requirements for the competence of testing and calibration laboratories*.

[2] ISO 9000:2005, *Quality management systems — Fundamentals and vocabulary*.

[3] JCGM 200:2012, *International vocabulary of metrology – Basic and general concepts and associated terms (VIM)*.

[4] Eurolab Technical Report No. 2/2006, 2006, *Guidance for the management of computers and software in laboratories with reference to ISO/IEC 17025/2005*.

[5] Gogates G D, 2012, *Software Validation in Accredited Laboratories A Practical Guide*, Fasor Inc. (www.fasor.com).

[6] Wichmann B, Parkin G and Barker R, 2007, *NPL Report DEM-ES 014 - Software Support for Metrology Best Practice Guide No. 1 - Validation of Software in Measurement Systems*.

[7] Torp C E, 2003, *Nordtest Report TR 535 - Method of Software Validation*.

[8] IEEE Std 1012-2012, 2012, *IEEE Standard for System and Software Verification and Validation.*

[9] ESA PSS-05-10, 1995, *Guide to software Verification and validation*, European Space Agency.

[10] ISO/IEC 12207:2008, *Systems and software engineering – Software life cycle processes*.

[11] ISO/IEC/IEEE 24765:2010, *Systems and software engineering – Vocabulary*.