

|  |
| --- |
| **First Committee Draft (1CD)**  Project: Revision of OIML D 31:2008  Title: General requirements for software controlled measuring instruments  Date: 10 January 2018  Document number: TC5\_SC2\_P3\_N019  Supersedes document: TC5\_SC2\_P3\_N016  Project Group: OIML TC 5/SC 2/p 3  Convenership: Germany  Conveners: Marko Esche + Federico Grasso Toro |
| Circulated to P- and O-members and liaison international bodies and external organizations for:  🞏 Discussion at (date and place of meeting):  ⌧ Comments by: 10 April 2018  🞏 Vote (P-members only) and comments by: |
|
|
|

List of Changes spread through the document:

New (basic) reference:

For several terms in the section “terms and definitions”, the reference document is changed to a document of more basic nature, e.g. OIML D 11 against OIML V 1, as long as the explanation roughly remains unchanged. In addition, all terms were updated to reflect requirements laid down in B 6-2:2012.

Less technical detail:

It seems to be appropriate to delete some of the technical details, since this is not a document explaining IT related features like CRC-32 or Elliptic curves.

At the Berlin meeting it was agreed to provide less technical detail in requirements. In examples, details may remain. See CA-2.

No extra declarations

The manufacturer is responsible for the correctness and completeness of the documentation. It brings no improvement for type evaluation, if the manufacturer is asked for extra papers like a declaration that the documented list of commands is complete.

Naming of hardware – measuring instrument

OIML B 3, VIM3 and OIML V 1 define a measuring instrument as a device used for measurements. It seems to be not necessary to define an electronic measuring instrument as a measuring instrument working electronically. If at all, a definition of a software-driven or software controlled measuring instrument would be more appropriate for this document.

Naming of hardware – device and electronic device

OIML B 3, VIM3 and OIML V 1 do not define the terms device and electronic device, but use them in a common sense. So, the terms device and electronic device should be replaced. All usages of device, electronic device, component etc. are replaced by sub-assembly or measuring instrument.

At the Berlin meeting it was agreed not to use the term "sub-assembly" any longer. Instead, "measuring instrument" or "component" are used. See NL-3, NL-32, NL-42, NL-51, JP-4, JP-2.

Use of the term "module" is avoided if hardware is meant. It is used for software parts.

"Sub-assembly" is still used in cited V 1-definitions.

**Contents**

*Foreword*

**1 Introduction 5**

**2 Scope and field of application 5**

**3 Terms and definitions 6**

3.1 General terminology 6

3.2 Abbreviations 13

**4 Instructions for use of this Document in drafting OIML Recommendations 13**

**5 Requirements for measuring instruments with respect to the application of software 13**

5.1 General requirements 13

5.2 Requirements specific for configurations 18

**6 Type evaluation 30**

6.1 Software documentation to be supplied for type evaluation 30

6.2 Requirements on the evaluation procedure 31

6.3 Validation methods (software examination) 32

6.4 Validation procedure 37

6.5 Equipment under test (EUT) 41

**7 Verification 41**

**8 Risk assessment 41**

Annex A Bibliography 43

Annex B Example of a software evaluation report 45

Annex C Index 52

**Foreword**

The International Organization of Legal Metrology (OIML) is a worldwide, intergovernmental organization whose primary aim is to harmonize the regulations and metrological controls applied by the national metrological services, or related organizations, of its Member States. The main categories of OIML publications are:

* **International Recommendations (OIML R),** which are model regulations that establish the metrological characteristics required of certain measuring instruments and which specify methods and equipment for checking their conformity. OIML Member States shall implement these Recommendations to the greatest possible extent;
* **International Documents (OIML D),** which are informative in nature and which are intended to harmonize and improve work in the field of legal metrology;
* **International Guides (OIML G),** which are also informative in nature and which are intended to give guidelines for the application of certain requirements to legal metrology; and
* **International Basic Publications (OIML B),** which define the operating rules of the various OIML structures and systems.

OIML Draft Recommendations, Documents and Guides are developed by Technical Committees or Subcommittees which comprise representatives from the Member States. Certain international and regional institutions also participate on a consultation basis. Cooperative agreements have been established between the OIML and certain institutions, such as ISO and the IEC, with the objective of avoiding contradictory requirements. Consequently, manufacturers and users of measuring instruments, test laboratories, etc. may simultaneously apply OIML publications and those of other institutions.

International Recommendations, Documents, Guides and Basic Publications are published in English (E) and translated into French (F) and are subject to periodic revision.

Additionally, the OIML publishes or participates in the publication of **Vocabularies (OIML V)** and periodically commissions legal metrology experts to write **Expert Reports (OIML E)**. Expert Reports are intended to provide information and advice, and are written solely from the viewpoint of their author, without the involvement of a Technical Committee or Subcommittee, nor that of the CIML. Thus, they do not necessarily represent the views of the OIML.

This publication - reference OIML D 31, edition YYYY (E) - was developed by the OIML Technical Subcommittee TC 5/SC 2 *Software*. It was approved for final publication by the International Committee of Legal Metrology in YYYY.

OIML Publications may be downloaded from the OIML web site in the form of PDF files. Additional information on OIML Publications may be obtained from the Organization’s headquarters:

Bureau International de Métrologie Légale

11, rue Turgot - 75009 Paris - France

Telephone: 33 (0)1 48 78 12 82

Fax: 33 (0)1 42 82 17 27

E-mail: [biml@oiml.org](mailto:biml@oiml.org)

Internet: [www.oiml.org](http://www.oiml.org/)

**General requirements**

**for software controlled measuring instruments**

# Introduction

The primary aim of this International Document is to provide OIML Technical Committees and Subcommittees with guidance for establishing appropriate requirements for software related functionalities in measuring instruments covered by OIML Recommendations.

Furthermore, this International Document can provide guidance to OIML Member States in the implementation of OIML Recommendations in their national laws.

# Scope and field of application

**2.1** This International Document specifies the general requirements applicable to legally relevant software related functionality and security in measuring instruments and gives guidance for verifying the compliance of an instrument with these requirements.

**2.2** This Document shall be taken into consideration by the OIML Technical Committees and Subcommittees as a basis for establishing specific software requirements and procedures in OIML Recommendations applicable to particular categories of measuring instruments (hereafter termed “relevant OIML Recommendations”).

**2.3** The instructions given in this Document apply only to software controlled measuring instruments or their components.

*Notes:*

* This Document does not cover all the technical requirements specific to software controlled measuring instruments; these requirements are to be given in the relevant OIML Recommendation, e.g. for weighing instruments, water meters, etc.
* This Document addresses some aspects concerning data security. In addition, national regulations for this area have to be considered.

# Terms and Definitions

Some of the definitions used in this Document are in conformity with the International Vocabulary of Metrology - Basic and General Concepts and Associated Terms 3rd Edition (OIML V 2-200:2012 [1]), with the International Vocabulary of Terms in Legal Metrology (OIML V 1:2013 [8]), with the OIML International Document *General requirements for measuring instruments – Environmental conditions* (OIML D 11:2013 [3]) and several ISO/IEC International Standards. For the purpose of this Document, the following definitions and abbreviations apply.

## General terminology

3.1.1

**audit trail**

continuous data file containing a time stamped information record of events, e.g. changes in the values of the parameters of a measuring instrument or software updates, or other activities that are legally relevant and which may influence the metrological characteristics

[OIML V 1:2013, 6.05]

3.1.2

**authentication**

checking of the declared or alleged identity of a user, process, or measuring instrument

*Note:* This may be necessary when checking that downloaded software originates from the owner of the type evaluation certificate.

3.1.3

**authenticity**

result of the process of authentication (passed or failed)

3.1.4

**checking facility**

facility that is incorporated in a measuring instrument and which enables significant defect to be detected and acted upon

*Note:* “Acted upon” refers to any adequate response by the measuring instrument (luminous signal, acoustic signal, prevention of the measurement process, etc.).

[adopted from OIML V 1:2013, 5.07]

3.1.5

**communication interface**

part of an instrument that enables information to be passed between measuring instruments, components of measuring instruments or other external systems

*Note 1:* Communication interfaces can be wired, optical, radio, etc. and are usually designed to use a specific protocol.

*Note 2:* This definition does not include communication between software parts.

3.1.6

**cryptographic certificate**

dataset containing the public key belonging to a measuring instrument or a person plus a unique identification of the subject, e.g. serial number of the measuring instrument or name or Personal Identification Number (PIN) of the person, plus a date of expiry

3.1.7

**cryptographic means**

software means like encryption/decryption with the purpose of hiding information from unauthorized persons, cryptographic hashes, or electronic signatures (see 3.1.12)

3.1.8

**data domain**

location in memory that each program needs for processing data

*Note:* Data domains may belong to one *software module* only, or to several.

3.1.9

**device-specific parameter**

legally relevant parameter with a value that depends on the individual instrument

*Note:* Device-specific parameters comprise adjustment parameters (e.g. span adjustment or other adjustments or corrections) and configuration parameters (e.g. maximum value, minimum value, units of measurement, etc.).

[OIML V 1:2013, 4.12]

3.1.10

**durability**

ability of the measuring instrument to maintain its performance characteristics over a period of use

[OIML V 1:2013, 5.15]

3.1.11

**electronic measuring instrument**

measuring instrument intended to measure an electrical or non-electrical quantity using electronic means and/or equipped with electronic parts

*Note:* For the purpose of this Document, auxiliary equipment, provided that it is subject to metrological control, is considered to be part of the measuring instrument.

[OIML D 11:2013, 3.1]

3.1.12

**electronic Signature**

software means which is added to software or data with the purpose to verify the origin of software or data, i.e. to prove their authenticity, or to check that the software or data are unchanged, i.e. to prove their integrity

*Note:* For electronic signing, a public key system is used in general, i.e. a pair of keys where only one has to be kept secret; the other may be public.

The secret key is used when software or data are secured. The public key is used when software are data are validated before use.

The validating instance may require a cryptographic certificate of the securing instance (see 3.1.6) to be sure of the authenticity of the public key.

3.1.13

**error of indication**

indication minus a reference quantity value

[OIML V 1:2013, 0.04]

3.1.14

**error log**

continuous data file containing an information record of failures/faults that have an influence on the metrological characteristics

3.1.15

**event**

action in which a modification of a measuring instrument parameter, adjustment factor or update of software module is made

[OIML V 1:2013, 6.06]

3.1.16

**event counter**

non-resettable counter that increments each time an event occurs

3.1.17

**executable code**

file installed on the computer system of the measuring instrument or component (EPROM, hard disk, etc.). This code is interpreted by the microprocessor and transposed into certain logical, arithmetical, decoding, or data transporting operations

3.1.18

**fault**

difference between the error of indication and the intrinsic error of a measuring instrument

*Note 1:* Principally, a fault is the result of an undesired change of data contained in or flowing through an electronic measuring instrument.

*Note 2:* From the definition it follows that a "fault" is a numerical value which is expressed either in a unit of measurement or as a relative value, for instance as a percentage.

[OIML V 1:2013, 5.12]

3.1.19

**hash function**

(mathematical) function which maps values from a large (possibly very large) domain into a smaller range. *Note:* A “good” hash function is such that the results of applying the function to a (large) set of values in the domain will be evenly distributed (and apparently at random) over the range.

[ISO/IEC 9594-8:2014] [4]

3.1.20

**integrity of programs, data, or parameters**

assurance that the programs, data, or parameters have not been subjected to any unauthorized or unintended changes while in use, transfer, storage, repair or maintenance

3.1.21

**interface**

shared boundary between two functional units, defined by various characteristics pertaining to the functions, physical interconnections, signal exchanges, and other characteristics of the units, as appropriate.

[ISO 2382-9:1995] [5]

3.1.22

**interruptible cumulative measurement**

process of cumulative measurement of the quantity value of a substance that can be easily and rapidly stopped during normal operation

*Note:* See also non-interruptible cumulative measurement (3.1.29).

3.1.23

**intrinsic error**

error of indication, determined under reference conditions

[OIML V 1:2013, 0.06]

3.1.24

**legally relevant**

subject to legal control

[to be changed in OIML V 1:2013, 4.08]

3.1.25

**legally relevant parameter**

parameter of a measuring instrument, (electronic) device, sub-assembly, software or a module subject to legal control

*Note:* The following types of legally relevant parameters can be distinguished: *type-specific parameters* and *device-specific parameters*.

[OIML V 1:2013, 4.10]

3.1.26

**legally relevant software part**

part of all *software modules* of a measuring instrument that is legally relevant

3.1.27

m**aximum permissible error (of a measuring instrument)**

extreme value of a measurement error, with respect to a known reference quantity value, permitted by specifications or regulations for a given measurement, measuring instrument, or measuring system

[OIML V 1:2013, 0.05]

3.1.28

**measuring instrument**

device used for making measurements, alone or in conjunction with one or more supplementary devices

[OIML V 1:2013, 0.10]

3.1.29

**non-interruptible cumulative measurement**

cumulative measuring process with no definite end that cannot be stopped and continued again by a user or an operator without falsifying the result of the measurement

*Note:* See also interruptible cumulative measurement (3.1.22)

3.1.30

**performance test**

test intended to verify whether the EUT is able to accomplish its intended functions

[OIML V 1:2013, 5.21]

3.1.31

**sealing**

means intended to protect the measuring instrument against any unauthorized modification, readjustment, removal of parts, software, etc.

*Note:* This may be achieved by hardware, software or a combination of both.asdf

[OIML V 1:2013, 2.20]

3.1.32

**securing**

means preventing unauthorized access to hardware or software

[OIML V 1:2013, 2.21]

3.1.33

**significant defect**

event that has an impact on the properties or functions of the measuring instrument or a fault

3.1.34

**software examination**

technical operation that consists of determining one or more characteristics of the software according to the specific procedure (e.g. analysis of technical documentation or running the program under controlled conditions)

3.1.35

**software identification**

sequence of readable characters (e.g. version number, checksum) that represents the software or software module under consideration.

*Note:* It can be checked on an instrument whilst in use.

3.1.36

**software interface**

program code and dedicated data domain; receiving, filtering, or transmitting data between *software modules*

*Note 1:* A software interface is not necessarily legally relevant.

*Note 2*: A software interface is an interface between two or more software modules, used to exchange data and transmit commands.

[OIML V 1:2013, 6.03]

3.1.37

**software module**

logic software entity such as a program, subroutine, library, parameter or data set, and other object including their *data domains* that may be in relationship with other entities

*Note:* The software of measuring instruments consists of one or more software modules.

3.1.38

**software protection**

protection of measuring instrument software or data domain by a hardware or software implemented seal

*Note:* The seal must be removed, damaged or broken to obtain access to change software.

[OIML V 1:2013, 6.04]

3.1.39

**software separation**

separation of the software in measuring instruments, which can be divided into a *legally relevant part* and a legally non-relevant part

*Note:* These parts communicate via a software interface.

[OIML V 1:2013, 6.02]

3.1.40

**source code**

computer program written in a form (programming language) that is legible and editable. Source code is compiled or interpreted into *executable code*

3.1.41

**storage device**

device used for storing measurement data after completion of the measurement and keeping it available for later legally relevant purposes (e.g. the conclusion of a commercial transaction)

[OIML V 1:2013, 6.07]

3.1.42

**time stamp**

unique monotonically increasing time value, e.g. in seconds or a date and time string denoting the date and/or time at which a certain event or fault occurred

3.1.43

**transmission of measurement data**

transmission of measurement data via communication lines or other means to a receiver where they are further processed

3.1.44

**type (pattern) evaluation**

conformity assessment procedure on one or more specimens of an identified type (pattern) of measuring instruments which results in an evaluation report / or an evaluation certificate

[OIML V 1:2013, 2.04]

3.1.45

**type-specific parameter**

*legally relevant parameter* with a value that depends on the type of instrument only

*Note:* Type-specific parameters are part of the legally relevant software.

[OIML V 1:2013, 4.11]

Example: Considering a measuring instrument of liquids other than water, the range of cinematic viscosity of a turbine is a type-specific parameter, fixed by the type evaluation of the turbine. All the manufactured turbines of the same type use the same range of viscosity.

3.1.46

**universal computer**

computer that is not constructed for a specific purpose but that can be adapted to the metrological task by software

*Note:* In general, this software is founded on an operating system that permits loading and execution of software for specific purposes.

3.1.47

**user interface**

interface that enables information to be interchanged between the operator and the measuring instrument or its hardware or software components

*Note:* Examples are switches, keyboard, mouse, display, monitor, printer, touch-screen, software window on a screen including the software that generates it.

[to be changed in OIML V 1:2013, 6.08]

3.1.48

**validation**

verification, where the specified requirements are adequate for an intended use

[OIML V 2-200:2012, 2.45]

3.1.49

**verification**

provision of objective evidence that a given item fulfils specified requirements

[OIML V 2-200:2012, 2.44]

## Abbreviations

EUT Equipment Under Test

IEC International Electrotechnical Commission

I/O Input / Output (refers to ports)

ISO International Organization for Standardization

IT Information Technology

MPE Maximum Permissible Error

OIML International Organization of Legal Metrology

PCB Printed Circuit Board

PIN Personal Identification Number

TEC Type evaluation certificate

# Instructions for use of this Document in drafting OIML Recommendations

4.1The provisions of this Document apply only to new OIML Recommendations and OIML Documents under revision. The OIML project groups (Technical Committees, Subcommittees) should use this guidance Document to establish software related requirements in addition to the other technical and metrological requirements of the applicable OIML Recommendation.

4.2All referred documents are subject to revision, and the users of this Document are encouraged to investigate the possibility of applying the most recent editions of the referred documents.

4.3It is the objective of this Document to provide the project groups responsible for drawing up OIML Recommendations with a set of requirements – partly with different levels – that are suitable to cover the demands of all kinds of measuring instruments and all areas of application. The project group shall determine which risk level is suitable, and how to incorporate the relevant portions of this Document into the OIML Recommendation being drafted. In Clause 8 some aid is given for performing this task.

# Requirements for measuring instruments with respect to the application of software

## General requirements

At the time of publishing this Document the general requirements represent the state of the art in information technology (IT). They are in principle applicable to all kinds of software controlled measuring instrumentscomponents of measuring instruments, and should be considered in all OIML Recommendations. In contrast to these general requirements the requirements specific for configuration (5.2) deal with technical features that are not common for some kinds of instruments or in some areas of application.

In the examples, where applicable, both normal and raised risk levels are shown. Notation in this

Document is as follows:

(I) Technical solution acceptable in case of normal risk level;

(II) Technical solution acceptable in case of raised risk level (see 8).

5.1.1 Software identification

Software of a measuring instrument/component shall be clearly identified. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.

The identification shall be displayed or printed:

* on command or
* during operation or
* at start up for a measuring instrument that can be turned off and on again.

If a measuring instrument or component has neither display nor printer, the identification shall be sent via a communication interface, in order to be displayed/printed on another component.

As an exception, an imprint of the software identification on the instrument/component shall be an acceptable solution if it satisfies the following conditions:

(1) The user interface does not have any control capability to activate the indication of the software identification on the display, or the display does not technically allow the identification of the software to be shown (analog indicating device or electromechanical counter).

(2) The instrument/component does not have an interface to communicate the software identification.

(3) After production of the instrument/component a change of the software is not possible, or only possible if the hardware is also changed.

It shall be ensured that the software identification is correctly marked on the concerned instrument/component.

The relevant OIML Recommendation should allow or disallow this exception.

If the software is modified in any way, a new software identification is required.

The software identification and the means of identification (e.g. software version, hash value, checksum) shall be stated in the type evaluation certificate. Instructions on how to display the software identification shall be in the TEC.

*Note:* Each measuring instrument in use has to conform to the approved type. The software identification enables surveillance personnel and persons affected by the measurement to determine whether the instrument under consideration is conformable.

Example:

The software contains a textual string or a number, unambiguously identifying the installed version. This string is transferred to the display of the instrument when a button is pressed, when the instrument is switched on, or cyclically controlled by a timer.

A version number may have the following structure A.Y.Z. Considering a flow computer; the letter A will represent the version of the core software that is counting pulses; the letter Y will represent the version of the conversion function (none, at 15 °C, at 20 °C); the letter Z will represent the language of the user interface.

5.1.2 Correctness of algorithms and functions

The measuring algorithms and functions of a measuring instrument shall be appropriate and functionally correct for the given application and device type (accuracy of the algorithms, price calculation according to certain rules, rounding algorithms, etc.).

The measurement result and accompanying information required by specific OIML Recommendations or by national legislation shall be displayed or printed correctly.

It shall be possible to examine algorithms and functions either by metrological tests, software tests or software examination (as described in 6.3).

No hidden or undocumented functions or parameters shall exist.

5.1.3 Software protection

5.1.3.1 Prevention of misuse

A measuring instrument shall be constructed in such a way that possibilities for unintentional, accidental, or intentional misuse are minimal. In the framework of this OIML Document, this applies especially to the software. The presentation of the measurement results should be unambiguous for all parties affected.

*Note:* Software controlled instruments are often complex in their functionality. The user needs good guidance for correct use and for achieving correct measurement results*.*

Example:

The user is guided by menus. The legally relevant functions are combined into one branch in this menu. If any measurement values might be lost by an action, the user is warned and requested to perform another action before the function is executed. See also 5.2.2.

5.1.3.2 Fraud protection

5.1.3.2.a Software shall be protected in such a way that evidence of any intervention (e.g. software updates, parameters changes) shall be available. Software shall be secured against unauthorized modification, loading, or changes by swapping the memory device. Mechanical sealing or other technical means may be necessary to secure measuring instruments.

Example:

(I)/(II) The housing containing the memory devices is sealed or the memory device is sealed on the PCB.

(II) If a rewritable device is used, the write-enable input is inhibited by a switch that can be sealed. The circuit is designed in such a way that the write protection cannot be cancelled by a short-circuit of contacts.

(I) A measuring instrument consists of two components, one containing the main metrological functions incorporated in a housing that is sealed. The other component is a universal computer with an operating system. Some functions such as the indication are located in the software of this computer. To prevent swapping of the software on the universal computer the data transfer between the component and the universal computer is encrypted. The key for decryption is hidden in the legally relevant program of the universal computer. Only this program knows the key and is able to read, decrypt and use the measurement values. Other programs cannot be used for this purpose as they cannot decrypt the measurement values (see also example in 5.2.1.2.d).

5.1.3.2.b Only clearly documented functions (see 6.1) may be activated by the user interface, which do not influence the metrological characteristics of the instrument.

*Note:* The examiner decides whether all of these documented commands are acceptable.

Example:

(I)/(II) All inputs from the user interface are redirected to a program that filters incoming commands. It only allows the documented ones and discards all others. This program is part of the legally relevant software.

5.1.3.2.c Parameters that fix the legally relevant characteristics of the measuring instrument shall be secured against unauthorized modification. If necessary for the purpose of verification, displaying or printing of the current parameter settings shall be possible.

*Note:* Some of thedevice-specific parameters may stay adjustable or selectable after type evaluation. They should be adjustable/selectable only in a special operational mode of the instrument.

The device-specific parameters may be classified as those that should be secured (unalterable) and those that may be accessed (adjustable/selectable parameters) by an authorized person, e.g. the instrument owner or product vendor.

Type-specific parameters have identical values for all specimens of a type. They are fixed at type evaluation of the instrument.

Example:

(I)/(II) Device specific parameters to be secured are stored in a non-volatile memory. The write-enable input of the memory is inhibited by a switch that is sealed.

Refer to examples 5.1.3.2.d (1) to (3) in this clause.

5.1.3.2.d Software protection comprises appropriate sealing by mechanical, electronic and/or cryptographic means, making an unauthorized intervention impossible or evident.

*Note*: A cryptographic certificate may be used. The software is signed by a trustworthy institution with an electronic signature. The assignment of a public key to a subject can be verified by using the public key of the trustworthy institution and decrypting the signature of the certificate.

Example:

(1) (I) Electronic sealing. The legally relevant parameters of an instrument can be input and adjusted by a menu item. The software recognizes each change and increments an event counter with each event of this kind. This event counter value can be indicated. The initial value of the event counter has to be registered in the TEC. If the indicated value differs from the registered one, the instrument is in an unverified state (equivalent to a broken seal).

(2) (I)/(II) The software of a measuring instrument is constructed such (see Example 5.1.3.2.a) that there is no way to modify the legally relevant parameters but via a switch protected menu. This switch is mechanically sealed in the inactive position, making modification of the legally relevant parameters impossible.

To modify the legally relevant parameters, the switch has to be activated, inevitably breaking the seal by doing so.

(3) (II) The software of a measuring instrument is constructed such that there is no way to access the legally relevant parameters but by authorized persons. If a person wants to access the parameter menu item, that person has to insert his smart card containing a PIN as part of a cryptographic certificate. The software of the instrument is able to verify the authenticity of the PIN by the certificate and allows the parameter menu item to be entered. The access is recorded in an audit trail including the identity of the person (or at least of the smart card used).

5.1.4 Support of hardware features

5.1.4.1 Support of fault detection

The relevant OIML Recommendation may require fault detection functions for certain faults of the instrument (addressed in OIML D 11:2013 (5.1.2 (b) and 5.3)). In this case, the manufacturer of the instrument shall be required to design checking facilities into the software or hardware parts or provide means by which the hardware parts can be supported by the software parts of the instrument.

If software is involved in fault detection, an appropriate reaction is required. The relevant OIML Recommendation may prescribe that the instrument / component is deactivated or an alarm / record in an error log is generated in case a fault condition is detected.

The documentation to be submitted for type evaluation shall contain a list of the faults that will be detected by the software and the expected reaction and in case needed for understanding its operation, a description of the detecting algorithm.

Example:

(I) On each start-up the legally relevant program calculates a checksum of the program code and legally relevant parameters. The nominal value of these checksums has been calculated in advance and stored in the instrument. If the calculated and stored values do not match, the program stops execution.

If the measurement is cumulative and not interruptible, the checksum is calculated cyclically and controlled by a software timer. In case a failure is detected, the software displays an error message or switches on a failure indicator and records the time of the fault in an error log (if one exists).

(II) On each start-up, the legally relevant program calculates a cryptographic hash of the program code and legally relevant parameters. The nominal value of the hash has been calculated in advance and stored in the instrument. If the calculated and stored values do not match, the program stops execution.

If the measurement is cumulative and not interruptible, the hash is calculated cyclically and controlled by a software timer. In case a failure is detected, the software displays an error message or switches on a failure indicator and records the time of the fault in an error log (if one exists).

5.1.4.2 Support of durability protection

It is the manufacturer’s choice to realize durability protection facilities addressed in OIML D 11:2013 (5.1.3 (b) and 5.4) in software or hardware, or to allow hardware facilities to be supported by software. The relevant OIML Recommendation may suggest appropriate solutions.

If software is involved in durability protection, an appropriate reaction is required. The relevant OIML Recommendation may prescribe that the instrument / component is deactivated or an alarm / report is generated in case durability is detected as being jeopardized.

Example:

(I)/(II) Some kind of measuring instruments require an adjustment after a prescribed time interval, in order to guarantee the durability of the measurement. The software gives a warning when the maintenance interval has elapsed and even stops measuring, if it has been exceeded for a certain time interval.

**5.2** Requirements specific for configurations

The requirements given in this clause are based on typical technical solutions in IT, although they might not be common in all areas of legal applications. Following these requirements technical solutions are possible that show the same degree of security and conformity to a type as instruments that are not software controlled.

The following specific requirements are needed when certain technologies are employed in measuring instruments. They have to be considered in addition to those described in 5.1.

In the examples, where applicable, both normal and raised risk levels are shown. Notation in this

document is as follows:

(I) Technical solution acceptable in case of normal risk level;

(II) Technical solution acceptable in case of raised risk level (see 8).

5.2.1 Specification and separation of legally relevant parts and specification of interfaces

This requirement applies if the measuring instrument (component) has interfaces for communicating with other instruments or components, with the user, or with other software parts besides the legally relevant parts within a measuring instrument (component).

Legally relevant parts of a measuring instrument – whether software or hardware parts – shall not be inadmissibly influenced by other parts of the measuring instrument.

OIML Recommendation may specify the software / hardware / data or part of the software/hardware/data that are legally relevant.

5.2.1.1 Separation of components

5.2.1.1.a Components of a measuring instrument that perform legally relevant functions shall be identified, clearly defined and documented. They form the legally relevant part of the measuring instrument.

*Note:* The examiner decides whether this part is complete and whether other parts of the measuring instrument may be excluded from further evaluation.

Example:

(1) (I)/(II) An electricity meter is equipped with an optical interface for connecting an electronic device to read out measurement values. The meter stores all the relevant quantities and keeps the values available for being read out for a sufficient time span. In this system only the electricity meter is the legally relevant instrument Other legally non-relevant devices may exist and may be connected to the interface that complies with clause 5.2.1.1.b. Securing of the data transmission itself (see 5.2.4) is not required.

(2) (I)/(II) A measuring instrument consists of the following components:

* a digital sensor calculating the weight or volume;
* a universal computer calculating the price;
* a printer printing out the measurement value and the price to pay.

All components are connected by a local area network. In this case the digital sensor, the universal computer and the printer are legally relevant components and are optionally connected to a merchandize system that is not legally relevant. The legally relevant components have to fulfill requirement 5.2.1.1.b and – because of the transmission via the network – also requirements contained in 5.2.4.

5.2.1.1.b It shall be demonstrated that the functions and data of components, that are legally relevant, cannot be inadmissibly influenced by commands received via the interface to the other, legally non-relevant parts.

This implies that there is an unambiguous assignment of each command to all initiated functions or data changes in the component.

*Note:* If “legally relevant” components interact with other “legally relevant” components, refer to 5.2.4.

Example:

(1) (I)/(II) The software of the electricity meter (see example (1) of 5.2.1.1.a above) is able to receive commands for selecting the quantities required. It combines the measurement value with additional information – e.g. time stamp, unit – and sends this dataset back to the requesting device. The software only accepts commands for the selection of valid allowed quantities and discards any other command, sending back only an error message. There may be securing means for the contents of the dataset but they are not required, as the transmitted dataset is not subject to legal control.

(2) (I)/(II) Inside the housing that is sealed there is a switch that defines the operating mode of the electricity meter: one switch setting indicates the secured mode and the other the free mode (securing means other than a mechanical seal are possible; see examples 5.1.3.2.a/.d). When interpreting received commands, the software checks the position of the switch: in the free mode, the command set that the software accepts is extended compared to the secured mode (e.g. it may be possible to adjust the calibration factor by a command that is discarded in the secured mode).

5.2.1.2 Specification and separation of software parts

5.2.1.2.a All software modules (programs, subroutines, objects, etc.) that perform legally relevant functions or that process legally relevant data form the legally relevant software part of a measuring instrument (or component). The conformity requirement applies to this part and it shall be made identifiable as described in 5.1.1.

If the separation of the software is not possible or needed, the software is legally relevant as a whole.

Example:

(I) A measuring instrument consists of several digital sensors connected to a personal computer that displays the measurement values. The legally relevant software on the personal computer is separated from the legally non-relevant parts by compiling all procedures realizing legally relevant functions (including presentation of results) into a dynamically linkable library. One or several legally non-relevant applications may call program procedures in this library. These procedures receive the measurement data from the digital sensors, calculate the measurement result, and display it in a software window.

5.2.1.2.b If the legally relevant software part communicates with other software parts, a software interface shall be defined. All communication shall be performed exclusively via this interface. The legally relevant software part and the interface shall be clearly documented. All legally relevant functions and data domains of the software shall be described to enable a type evaluation authority to decide on correct software separation.

The interface consists of program code and dedicated data domains. Defined coded commands or data are exchanged between the software parts by storing to the dedicated data domain by one software part and reading from it by the other. Writing and reading program code is part of the software interface. The declared software interface shall not be circumvented.

5.2.1.2.c There shall be an unambiguous assignment of each command to all initiated functions or data changes in the legally relevant part of the software. Commands that communicate through the software interface shall be declared and documented. Only documented commands are allowed to be activated through the software interface.

Example:

(I) In the example described in 5.2.1.2.a the software interface consists of the procedures in the library and their parameters and return values. The interface cannot be circumvented e.g. by pointers to internal data. The number and kind of procedures, parameters, and return values is fixed at compile time.

(II) Legally relevant and legally non-relevant software run in separate virtual machines on a universal computer. Both machines are configured in such a way, that any communication between both software parts can only be done via the defined software interface. The setup of the virtual machines, including the method of communication between both, is part of the legally relevant software. The operating system ensures that the configuration cannot be modified without breaking a seal.

5.2.1.2.d Where legally relevant software has been separated from non-relevant software, the legally relevant software shall have priority using the resources over non-relevant software. The legally relevant process should not be interrupted by legally non-relevant software. The measurement process (realized by the legally relevant software part) must not be delayed or blocked by other processes.

Examples:

(1) (I) In the example 5.2.1.2.a/.c the legally non-relevant application controls the start of the legally relevant procedures in the library. Omitting a call of these procedures would of course inhibit the legally relevant function of the system. Therefore, the following provisions have been made in the example system to fulfill the requirement 5.2.1.2.d: The digital sensors send the measurement data in encrypted form. The key for decryption is hidden in the library. Only the procedures in the library know the key and are able to read, decrypt, and display measurement values.

(2) (I) The software of an electronic electricity meter reads raw measurement values from an analog-digital converter (ADC). For the correct calculation of the measurement values the delay between the “data ready” event from the ADC to finishing buffering of the measurement values is crucial. The raw values are read by an interrupt routine initiated by the “data ready” signal. The instrument is able to communicate via an interface with other electronic devices in parallel served by another interrupt routine (legally non-relevant communication). The priority of the interrupt routine for processing the measurement values is higher than that of the communication routine.

(II) Legally relevant and legally non-relevant software run in separate virtual machines on a universal computer. The configuration of the operating system ensures, that the virtual machine, on which the legally relevant software runs, has always sufficient system resources available for the legally relevant processes.

Examples from 5.2.1.2.a, 5.2.1.2.c and 5.2.1.2.d (1) are acceptable as a technical solution only for a normal risk level (I). If increased protection against fraud or increased conformity is necessary (see 8), software separation alone is not sufficient and additional means are demanded or the whole software should be considered as under legal control.

5.2.2 Shared indications

A display or printout may be employed for presenting both information from the legally relevant software part and other information. The contents and layout are specific for the kind of instrument and area of application and have to be defined in the relevant Recommendation. If a display or printout is used both for legally relevant and legally non-relevant outputs, the legally relevant information should always be readable, and clearly distinguishable from other information.

Example:

(I) In the measuring instrument described in the examples 5.2.1.2.a to 5.2.1.2.d, the measurement values are displayed in a separate software window. The means described in 5.2.1.2.d guarantee that only the legally relevant software can read and display the measurement values. The instrument has an operating system with a multiple windows user interface. The window displaying the legally relevant data is generated and controlled by procedures in the legally relevant dynamically linkable library (see 5.2.1.2). During measurement, these procedures check cyclically that the relevant window is still on top of all the other open windows; if not, the procedures put it on top.

(II) In the measuring instrument described in the examples 5.2.1.2.a to 5.2.1.2.d the measurement application runs in kiosk mode. The entire display is controlled by the legally relevant software. Legally non-relevant data is presented in a special part of the display marked as legally non-relevant.

If increased protection against fraud is necessary (II), a printout as an indication alone may not be suitable. There should exist a component with increased securing means that is able to display the measurement values.

The use of a universal computer is not appropriate as part of a measuring instrument if increased protection against fraud is necessary (II). Additional precautions to prevent or minimize the risk of fraud, in the form of hardware and software, should be considered when increased protection is necessary, such as when using a universal computer (for example PC, PDA, etc.).

5.2.3 Storage of data

If measurement values are stored before they are used for legal purposes the following requirements apply:

5.2.3.1 The measurement value stored shall be accompanied by all relevant information necessary for future legally relevant use.

Example:

(I)/(II) A dataset includes the following entries:

* measurement value including unit;
* time stamp of measurement (see 5.2.3.4);
* place of measurement or identification of the measuring instrument that was used for the measurement;
* unambiguous identification of the measurement, e.g. consecutive numbers enabling assignment to values printed on an invoice.

5.2.3.2 The stored data shall be protected by software means to guarantee the authenticity, integrity and, if necessary, correctness of the information concerning the time of measurement. The software that displays or further processes the measurement values and accompanying data shall check the time of measurement, authenticity, and integrity of the data after having read them from the storage. If an irregularity is detected, the data shall be discarded or marked unusable.

Software modules that prepare data for storing, or that check data after reading are considered part of the legally relevant software.

*Note:* It is appropriate to require a raised risk level when considering a freely accessible storage.

Raised risk levels might require the application of cryptographic methods. Means shall be provided whereby keys can only be input or read if a seal is broken.

Example:

(I) The program of the storing device calculates a CRC32 checksum of the dataset and appends it to the dataset. It uses a secret initial value for this calculation instead of the value given in the standard. This initial value is employed as a key and stored as a constant in the program code. The reading program also has stored this initial value in its program code. Before using the dataset, the reading program calculates the checksum and compares it with that stored in the dataset. If both values match, the dataset is not falsified. Otherwise, the program assumes falsification and discards the dataset.

Example:

(II) The storing program generates an electronic signature for the stored dataset. It is appended to the stored dataset. The private and public key used for signing are generated in a hardware security module which protects the private key against manipulation or reading and exports the public key. The reading program validates the signature with the public key to check authenticity and integrity of the dataset.. To prove the origin of the dataset the reading program must know whether the public key really belongs to the storing program. Therefore, the public key is presented on the display of the measuring instrument and can be registered once, e.g. together with the serial number of the instrument when it is legally verified in the field.

5.2.3.3 Automatic storing

5.2.3.3.a When, considering the application, data storage is required, measurement data must be stored automatically when the measurement is concluded, i.e. when the final value used for the legal purpose has been generated.

The storage device must have sufficient permanency to ensure that the data are not corrupted under normal storage conditions. There shall be sufficient memory storage for any particular application.

When the final value used for the legal purpose results from a calculation, all data that are necessary for the calculation must be automatically stored with the final value.

*Note:* In case of cumulative measurements it may happen that the same data domain (program variable) is used repeatedly. Questions of storage capacity may not play a role then.

5.2.3.3.b Stored data may be deleted if either:

* the transaction is settled;
* these data are printed by a printing device subject to legal control.

*Note:* Other general national regulations (e.g. for tax purposes) may contain strict limitations for the deletion of stored measurement data.

5.2.3.4 Time stamp

The time stamp is a date/time information in a consistent format, allowing for easy comparison of two different records and tracking progress over time.

The time stamp shall be read from the clock of the instrument. Depending on the kind of instrument, or area of application, setting the clock may be legally relevant and appropriate protection means shall be taken according to the risk level to be applied (see 5.1.3.2.c).

The internal clock of a stand-alone measuring instrument tends to have a large uncertainty because there is no means to synchronize it with the global clock. But if the information concerning the time of measurement is necessary for a specific field of application, the reliability of the internal clock of the measuring instrument shall be enhanced by specific means.

Example:

(II) The reliability of the internal quartz-controlled clock device of the measuring instrument is enhanced by redundancy: A timer is incremented by the clock of the microcontroller that is derived from another quartz crystal. When the timer value reaches a preset value, e.g. 1 second, a specific flag of the microcontroller is set and an interrupt routine of the program increments a second counter. At the end of e.g. one day the software reads the quartz-controlled clock device and calculates the difference in the seconds counted by the software. If the difference is within predefined limits, the software counter is reset and the procedure repeats; but if the difference exceeds the limits, the software initiates an appropriate error reaction.

5.2.4 Transmission via communication lines

If measurement values are transmitted before they are used for legal purposes the following requirements apply:

5.2.4.1 The measurement value transmitted shall be accompanied by all relevant information necessary for future legally relevant use.

Example:

(I)/(II) A dataset includes the following entries:

* measurement value including unit;
* time stamp of measurement (see 5.2.4.5);
* place of measurement or identification of the measuring instrument that was used for the measurement;
* unambiguous identification of the measurement, e.g. consecutive numbers enabling assignment to values printed on an invoice.

5.2.4.2 The transmitted data shall be protected by software means to guarantee the authenticity, integrity and, if necessary correctness of the information concerning the time of measurement. The software that displays or further processes the measurement values and accompanying data shall check the time of measurement, authenticity, and integrity of the data received them from a transmission channel. If an irregularity is detected, the data shall be discarded or marked unusable.

Software modules that prepare data for sending, or that check data after receiving, are considered part of the legally relevant software.

*Note:* It is appropriate to require a raised risk level when considering an open network.

Raised risk levels might require application of cryptographic methods. Means shall be provided whereby these keys can only be input or read if a seal is broken.

Example:

(I) The program of the sending device calculates a checksum of the dataset (algorithm such as BCC, CRC16, CRC32, etc.) and appends it to the dataset. It uses a secret initial value for this calculation instead of the value given in the standard. This initial value is employed as a key and stored as a constant in the program code. The receiving program also has stored this initial value in its program code. Before using the dataset, the receiving program calculates the checksum and compares it with that stored in the dataset. If both values match, the dataset is not falsified. Otherwise, the program assumes falsification and discards the dataset.

Example:

(II) The transmitting program generates an electronic signature for the transmitted dataset. It is appended to the transmitted dataset. The private and public key used for signing are is generated in a hardware security module which protects the private key against manipulation or reading and exports the public key. The receiving program validates the signature with the public key to check authenticity and integrity of the dataset.. To prove the origin of the dataset the receiving program must know whether the public key really belongs to the transmitting program. Therefore, the public key is presented on the display of the measuring instrument and can be registered once, e.g. together with the serial number of the instrument when it is legally verified in the field.

5.2.4.3 Transmission delay

The measurement shall not be inadmissibly influenced by a transmission delay.

5.2.4.4 Transmission interruption

If network services become unavailable, no measurement data shall be lost. The measurement process should be stopped to avoid the loss of measurement data.

*Notes:* Consideration should be given to distinguish between static and dynamic measurements.

Dependingon the area of application, and for cases where measurements are easily repeatable, a loss of transmitted data may be acceptable.

Example:

(I)/(II) The sending instrument/component waits until the receiver has sent an affirmation of correct receipt of the dataset. The sending instrument/component keeps the dataset in a buffer until this affirmation has been received. The buffer may have a capacity for more than one dataset, organized as a FIFO (First-in-first-out) queue.

5.2.4.5 Time stamp

The time stamp is a date/time information in a consistent format, allowing for easy comparison of two different records and tracking progress over time.

The time stamp shall be read from the clock of the instrument Depending on the kind of instrument, or area of application, setting the clock may be legally relevant and appropriate protection means shall be taken according to the risk level to be applied (see 5.1.3.2.c).

The internal clock of a stand-alone measuring instrument tends to have a large uncertainty because there is no means to synchronize it with the global clock. But if the information concerning the time of measurement is necessary for a specific field of application, the reliability of the internal clock of the measuring instrument shall be enhanced by specific means.

Example:

(II) The reliability of the internal quartz-controlled clock device of the measuring instrument is enhanced by redundancy: A timer is incremented by the clock of the microcontroller that is derived from another quartz crystal. When the timer value reaches a preset value, e.g. 1 second, a specific flag of the microcontroller is set and an interrupt routine of the program increments a second counter. At the end of e.g. one day the software reads the quartz-controlled clock device and calculates the difference in the seconds counted by the software. If the difference is within predefined limits, the software counter is reset and the procedure repeats; but if the difference exceeds the limits, the software initiates an appropriate error reaction.

5.2.5 Compatibility of operating systems and hardware

5.2.5.1 If an operating system is part of the measuring instrument, requirements according to this clause shall be met.

5.2.5.1.a The manufacturer shall state the legally relevant components of the operating system (e.g. libraries, drivers used in the measurement process, access management, security measures for interfaces). The protective features of software might rely on components of the operating system.

5.2.5.1.b The configuration of the operating system shall be identifiable.

*Note:* The access control might need to be configured in such way that the intended use of the measuring instrument cannot be inadmissibly influenced.

*Note 2:* Control of the data flow by the protection of interfaces (e.g. firewall configuration) shall be one of the main tasks in the configuration of a standard operating system.

5.2.5.1.c Changes to the configuration of the operating system shall be traceable.

5.2.5.2 The manufacturer shall identify the hardware and software environment that is suitable. Minimum resources and a suitable configuration (e.g. processor, memory, specific communication, version of operating system, etc.) necessary for correct functioning shall be declared by the manufacturer and stated in the type evaluation certificate.

5.2.5.3 Technical means shall be provided in the legally relevant software to prevent operation, if the minimal configuration requirements are not met. The system shall be operated only in the environment specified by the manufacturer for its correct functioning.

For example, in case an invariant environment is specified for the correct functioning of the system, means shall be provided to keep the operating environment fixed. This especially applies to a universal computer performing legally relevant functions.

Fixing the hardware, operating system, or system configuration of a universal computer or even excluding the usage of an off-the-shelf universal computer has to be considered in the following cases:

* if high conformity is required ;
* if cryptographic algorithms or keys have to be implemented (see 5.2.3 and 5.2.4).

5.2.6 Conformity of manufactured devices to the approved type

The manufacturer shall produce devices and legally relevant software that conform to the approved type and the documentation submitted.

5.2.7 Maintenance and re-configuration

Updating the legally relevant software of a measuring instrument in the field should be considered as:

* a modification of the measuring instrument, when exchanging the software with another approved version;
* a repair of the measuring instrument, when re-installing the same version.

A measuring instrument which has been modified or repaired while in service may require initial or subsequent verification, dependent on national regulations.

Software which does not realize legally relevant functions of the measuring instrument does not require verification after being updated.

5.2.7.1 Only versions of legally relevant software that conform to the approved type are allowed for use (see 5.2.6). They shall be stated in the TEC. Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. The following options 5.2.7.2 and 5.2.7.3 are alternatives. In case that device-specific parameters (especially calibration parameters) are concerned only verified update should be done.

This issue concerns verification in the field. Refer to Clause 7 for additional constraints.

5.2.7.2 Verified Update

The software to be updated can be loaded locally, i.e. directly on the measuring instrument or remotely via a network. Loading and installation may be two different steps (as shown in Fig. 1) or combined into one, depending on the needs of the technical solution. A physical seal needs to be broken for the update to take effect. A person should be on the installation site of the measuring instrument to check the effectiveness of the update. After the update of the legally relevant software of a measuring instrument (exchange with another approved version or re-installation) the measuring instrument is not allowed to be employed for legal purposes before a verification of the instrument as described in Clause 7 has been performed and the securing means have been renewed (if not otherwise stated in the relevant OIML Recommendation or in the TEC).

5.2.7.3 Traced Update

The software is implemented in the instrument according to the requirements for Traced Update (5.2.7.3.a to 5.2.7.3.h), if it is in compliance with the relevant OIML Recommendation. Traced Update is the procedure of changing software in a verified instrument or component after which the subsequent verification by a responsible person at place is not necessary. This means the traced update shall not affect existing parameters. The software to be updated can be loaded locally, i.e. directly on the measuring instrument or remotely via a network. The software update is recorded in an audit trail (see 3.1.1). The procedure of a Traced Update comprises several steps: loading, integrity checking, checking of the origin (authentication), installation, logging and activation.

5.2.7.3.a Traced Update of software shall be automatic. If some of the securing measures of the instrument are turned off to enable updating, they must be turned on again immediately after update, independent of the result of the update process.

5.2.7.3.b Software shall be protected in such a way that evidence of any intervention shall be available. During an update, any existing audit trail information shall be retained.

5.2.7.3.c Technical means shall be employed to guarantee the authenticity of the loaded software, i.e. that it originates from the owner of the type evaluation certificate.

Example:

(II) The authenticity check is accomplished by cryptographic means such as a public key system. The owner of the type evaluation certificate (in general the manufacturer of the measuring instrument) generates an electronic signature of the software to be updated using the *secret key* in the manufactory. The *public key* is stored in a fixed software part of the measuring instrument. The signature is checked using the *public key* when loading the software into the measuring instrument. If the signature of the loaded software is OK, it is installed and activated; if it fails the check, the fixed software discards it and uses the previous version of the software or switches to an inoperable mode.

5.2.7.3.d Technical means shall be employed to ensure the integrity of the loaded software, i.e. that it has not been inadmissibly changed before loading. This can be accomplished by adding a checksum or hash code of the loaded software and verifying it during the loading procedure.

5.2.7.3.e An audit trail shall be employed to ensure that Traced Updates of legally relevant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection.

The audit trail shall contain at minimum the following information: success / failure of the update procedure, software identification of the installed version, software identification of the previous installed version, time stamp of the event, identification of the downloading party if available An entry is generated for each update attempt regardless of the success.

The storage device that supports the Traced Update shall have a sufficient capacity to ensure the traceability of Traced Updates of legally relevant software between at least two successive verifications in the field/inspections. After having reached the limit of the storage for the audit trail, it shall be ensured by technical means that further downloads are impossible without breaking a seal.

The audit trail shall be displayed or printed on command. The TEC shall describe how the audit trail may be displayed or printed.

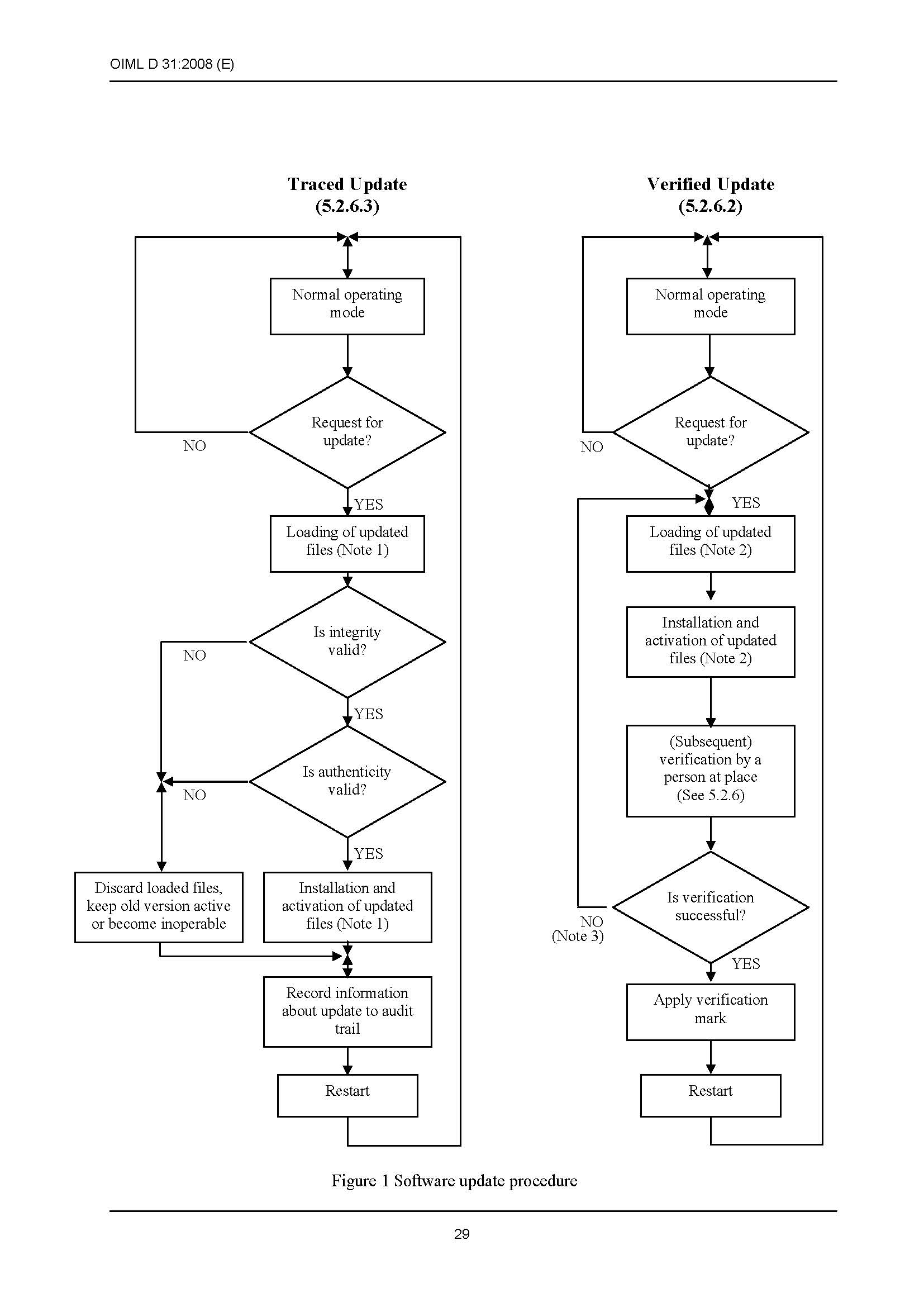
*Note:* This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace Traced Updates of legally relevant software over an adequate period of time (depending on national legislation).

5.2.7.3.f Depending on the needs and on national legal legislation it may be necessary for the user or owner of the measuring instrument to have to give his consent to a traced update. The measuring instrument shall have a feature for the user or owner to express his consent, e.g. a push button, before the update starts. It shall be possible to enable and disable the feature, e.g. by a switch that can be sealed or by a parameter. If the feature is enabled, each traced update has to be initiated by the user or owner. If it is disabled, no activity by the user or owner is necessary to perform a traced update.

5.2.7.3.g If the requirements in 5.2.7.3.a through 5.2.7.3.f cannot be fulfilled, it is still possible to update the legally non-relevant software part. In this case the following requirements shall be met:

* there is a distinct separation between the legally relevant and non-relevant software according to 5.2.1;
* the whole legally relevant software part cannot be updated without breaking a seal;
* it is stated in the type evaluation certificate that updating of the legally non-relevant part is acceptable.

5.2.7.3.h If the loaded software fails integrity test (5.2.7.3.d) or authenticity test (5.2.7.3.c), the instrument shall discard the new version and use the previous version of the software or switch to an inoperable mode. In this mode, the measuring functions shall be inhibited. It shall only be possible to resume the download procedure, or to show an error. If the audit trail has no more capacity (5.2.7.3.e), or the user or owner denies consent (5.2.7.3.f), the update procedure should not start at all.



*Notes:*  (1) In the case of a Traced Update, updating is separated into two steps: “loading” and “installing/activating”. This implies that the software is temporarily stored after loading without being activated because it must be possible to discard the loaded software and revert to the old version, if the checks fail.

(2) In the case of a Verified Update, the software may also be loaded and temporarily stored before installation but depending on the technical solution loading and installation may also be accomplished in one step.

(3) Here, only failure of the verification due to the software update is considered. Failure due to other reasons does not require re-loading and re-installing of the software, symbolized by the NO-branch.

5.2.7.4 The relevant OIML Recommendation may require the setting of certain device-specific parameters to be available to the user. In such a case, the measuring instrument shall be fitted with a facility to automatically and non-erasably record any adjustment of the device-specific parameter, e.g. an audit trail. The instrument shall be capable of presenting the recorded data.

5.2.7.5 The audit trails are part of the legally relevant software and should be protected as such. They should not be exchanged when the software is updated.

# Type evaluation

## Software documentation to be supplied for type evaluation

For type evaluation the manufacturer of the measuring instrument shall declare and document all program functions, relevant data structures and software interfaces of the legally relevant software part that are implemented in the instrument. All commands and their effects shall be described completely in the software documentation to be submitted for type evaluation.

Furthermore, the application for type evaluation shall be accompanied by a document or other evidence that supports the assumption that the design and characteristics of the software of the measuring instrument comply with the requirements of the relevant OIML Recommendation, in which the general requirements of this Document have been incorporated.

6.1.1 Typical documentation (for each measuring instrument, component) basically includes:

* a description of the legally relevant software and how the requirements are met:

- list of software modules that belong to the legally relevant part;

- description of the software interfaces of the legally relevant software part and of the commands and data flows via this interface;

- depending on the validation method chosen in the relevant OIML Recommendation (see 6.3 and 6.4) the source code shall be made available to the type evaluation authority if raised risk level is required by the relevant OIML Recommendation;

- list of parameters to be protected and description of protection means;

* a description of suitable system configuration and minimal required resources (see 5.2.5);
* a description of security means of the operating system (password, etc. if applicable);
* a description of the (software) sealing method(s);
* an overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc. Where a hardware component is deemed legally relevant or where it performs legally relevant functions, this should also be identified;
* a description of the accuracy of the algorithms (e.g. filtering of A/D conversion results, price calculation, rounding algorithms, etc.);
* a description of the user interface, menus and dialogues;
* the software identification and instructions for obtaining it from an instrument in use;
* list of commands of each hardware interface of the measuring instrument /component ;
* list of durability errors that are detected by the software and if necessary for understanding, a description of the detecting algorithms;
* a description of datasets stored or transmitted;
* if fault detection is realized in the software, a list of faults that are detected and a description of the detecting algorithm;
* if an audit trail is realized in the software, a description on how to access the audit trail;
* the operating manual.

## Requirements on the evaluation procedure

Test procedures in the framework of the type evaluation are based on well-defined test setups and test conditions and can rely on precise comparative measurements. “Testing” and “validating” software are different activities. The accuracy or correctness of software in general cannot be measured in a metrological sense, though there are standards that prescribe how to “measure” software quality [e.g. ISO/IEC 14598]. The procedures described here take into consideration both the legal metrology needs and also well-known validation and test methods in software engineering but which do not have the same goals (e.g. a software developer who searches for errors but who also optimizes performance). As shown in 6.4 each software requirement needs individual adaptation of suitable validation procedures. The effort for the procedure should reflect the risk level.

The aim is to validate the fact that the instrument to be approved complies with the requirements of the relevant OIML Recommendation. For software controlled instruments the validation procedure comprises examinations, analysis, and tests and the relevant OIML Recommendation shall include an appropriate selection of methods described below.

The methods described below focus on the type examination. Verifications of every single instrument in use in the field are not covered by those validation methods. Refer to Clause 7 *Verification* for more information.

The methods specified for software validation are described in 6.3. Combinations of these methods forming a complete validation procedure adapted to all requirements defined in Clause 5 are specified in 6.4.

The manufacturer shall attest that no hidden or undocumented properties exist. (e.g. parameters, commands, functions, backdoors.)

6.2.1 Information to be included in the TEC

* The software identification of all approved versions.
* The method to display the current software identification on the approved instrument in use.
* The securing means and the method to check them (e.g. hardware seals, event counters, audit trails.)

## Validation methods (software examination)

6.3.1 Overview of methods and their application

The selection and sequence of the following methods are not prescribed and may vary in a validation procedure from case to case.

This is a rough overview. For more details, see 6.3.2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Abbreviation** | **Description** | **Application** | **Preconditions, tools for**  **application** | **Special skills for performing** |
| AD | Analysis of the documentation and validation of the design (6.3.2.1) | Always | Documentation | - |
| VFTM | Validation by functional testing of metrological functions (6.3.2.2) | Correctness of the algorithms, uncertainty, compensating and correcting algorithms, rules for price calculation | Documentation, specimen | - |
| VFTSw | Validation by functional testing of software functions (6.3.2.3) | Correct functioning of communication, indication, fraud protection, protection against operating errors, protection of parameters, fault detection | Documentation, specimen | - |
| DFA | Metrological data flow analysis (6.3.2.4) | Software separation, evaluation of the impact of commands on the instrument’s functions | Source code, tools for analyzing source code | Knowledge of programming languages |
| CIWT | Code inspection and walkthrough (6.3.2.5) | All purposes | Source code, tools for analyzing source code | Knowledge of programming languages |
| SMT | Software module testing (6.3.2.6) | All purposes when input and output can clearly be defined | Source code, testing environment | Knowledge of programming languages |

Table 1: Overview of the proposed selected validation methods

6.3.2 Description of selected validation methods

6.3.2.1 Analysis of Documentation and Specification and Validation of the Design (AD)

Application:

Basic procedure for software validation.

Preconditions:

The procedure is based on the manufacturer’s documentation of the measuring instrument. Depending on the demands this documentation shall have adequate scope:

(1) Specification of the externally accessible functions of the instrument in a general form (Suitable for simple instruments with no interfaces except a display, all features verifiable by functional testing, low risk of fraud);

(2) Specification of software functions and interfaces (necessary for instruments with interfaces and for instrument functions that cannot be functionally tested and in case of increased risk of fraud). The description shall make evident and explain all software functions that may have an impact on metrological features;

(3) Concerning interfaces, the documentation shall include a complete list of commands or signals that the software is able to interpret. The effect of each command shall be documented in detail. The way in which the instrument reacts on undocumented commands shall be described;

(4) Additional documentation of the software for complex measuring algorithms, cryptographic functions, or crucial timing constraints shall be provided, if necessary for understanding and evaluating the software functions;

A general precondition for examination is the completeness of the documentation and the clear identification of the EUT, i.e. of the software packages that contribute to the metrological functions (see 6.1.1).

Description:

The examiner evaluates the functions and features of the measuring instrument using the documentation and decides whether they comply with the requirements of the relevant OIML Recommendation. Metrological requirements as well as software-functional requirements defined in Clause 5 (e.g. fraud protection, protection of adjustment parameters, disallowed functions, communication with other devices, update of software, fault detection, etc.) shall be considered and evaluated. This task may be supported by the Software Evaluation Report Format (see Annex B Checklist).

Result:

The procedure gives a result for all characteristics of the measuring instrument, provided that the appropriate documentation has been submitted by the manufacturer. The result should be documented in a clause related to software in a Software Evaluation Report (see Annex B) included in the Evaluation Report Format of the relevant OIML Recommendation.

Complementary procedures:

Additional procedures should be applied, if examining the documentation cannot provide substantiated validation results. In most cases “Validating the metrological functions by functional testing” (see 6.3.2.2) is a complementary procedure.

References:

IEC 61508-5, 2010 [9].5

6.3.2.2 Validation by Functional Testing of the Metrological Functions (VFTM)

Application:

For validating correctness of algorithms for calculating the measurement value from raw data, for linearization of a characteristic, compensation of environmental influences, rounding in price calculation, etc.

Preconditions:

Operating manual, functioning specimen, metrological references, test equipment, test cases, instructions for test equipment.

When it is not clear how to validate a function of a software program the onus to develop a test method should be placed on the manufacturer. In addition, the services of the programmer should be made available to the examiner for the purposes of answering questions.

Description:

Most of the evaluation and test methods described in OIML Recommendations are based on reference measurements under various conditions. Their application is not restricted to a certain technology of the instrument. Although it does not aim primarily at validating the software, the test result can be interpreted as a validation of some software parts, in general even the metrologically most important. If the tests described in the relevant OIML Recommendation cover all the metrologically relevant features of the instrument, the corresponding software parts can be regarded as being validated. In general, no additional software analysis or test has to be applied to validate the metrological features of the measuring instrument.

Result:

Algorithms are correct or not correct. Measurement values under all conditions are within the MPE or not.

Complementary procedures:

The method is normally an enhancement of 6.3.2.1. In certain cases, it may be easier or more effective to combine the method with examinations based on the source code (6.3.2.5) or by simulating input signals (6.3.2.6) e.g. for dynamic measurements.

References:

Various specific OIML Recommendations.

6.3.2.3 Validation by Functional Testing of the Software Functions (VFTSw)

Application:

For validation of e.g. protection of parameters, indication of a software identification, software supported fault detection, configuration of the system (especially of the software environment), etc.

Preconditions:

Operating manual, software documentation, functioning specimen, test equipment, test cases, instructions for test equipment.

When it is not clear how to validate a function of a software program the onus to develop a test method should be placed on the manufacturer. In addition, the services of the programmer should be made available to the examiner for the purposes of answering questions.

Description:

Required features described in the operating manual, instrument documentation or software documentation are checked practically. If they are software controlled, they are to be regarded as validated if they function correctly without any further software analysis. Features addressed here are e.g.:

* Normal operation of the instrument, if its operation is software controlled. All switches or keys and described combinations should be employed and the reaction of the instrument evaluated. In graphical user interfaces, all menus and other graphical elements should be activated and checked;
* Effectiveness of parameter protection may be checked by activating the protection means and trying to change a parameter;
* Effectiveness of the protection of stored data may be checked by changing some data in the file and then checking whether this is detected by the program;
* Indication of the software identification may be validated by practical checking;
* If fault detection is software supported, the relevant software parts may be validated by provoking, implementing or simulating a fault and checking the correct reaction of the instrument;
* Protection means can be checked by making unauthorized changes. The software should inhibit these changes or should cease to function.

Result:

Software controlled feature under consideration is OK or not OK.

Complementary procedures:

Some features or functions of a software controlled instrument cannot be practically validated as described. If the instrument has interfaces, it is in general not possible to detect unauthorized commands only by trying commands at random. Besides that, a sender is needed to generate these commands. For the normal validation level method 6.3.2.1 may cover this requirement. For the extended examination level, a software analysis such as 6.3.2.4 or 6.3.2.5 is necessary.

References:

WELMEC Guide 2.3 Section 3 [100]; WELMEC Guide 7.2 Sections 4.2 and 5.2[11].

6.3.2.4 Metrological Dataflow Analysis (DFA)

Application:

For analysis of the software design concerning the control of the data flow of measurement values through the data domains that are subject to legal control, including the examination of the software separation.

Preconditions:

Software documentation, source code, editor, text search program or special tools. Knowledge of programming languages.

Description:

It is the aim of this method to find all parts of the software that are involved in the calculation of the measurement value or that may have an impact on it. Starting from the hardware port where measurement raw data from the sensor are available, the subroutine that reads them is searched. This subroutine will store them in a variable after possibly having done some processing From this variable the intermediate value is read by another subroutine and so forth until the completed measurement value is output to the display. All variables that are used as storage for intermediate measurement values and all subroutines processing and transporting these values can be found in the source code simply by using a text editor and a text search program to find all other occurrences of variable or subroutine name.

Other data flows can be found by this method, e.g. from interfaces to the interpreter of received commands. Furthermore, circumvention of a software interface (see 5.2.1.2) can be detected.

Result:

It can be validated whether software separation according to 5.2.1.2 is OK or not OK.

It can be validated whether the documented list of commands for each interface is complete or not.

Complementary procedures:

This method is recommended if software separation is realized and if high conformity or strong protection against manipulation is required. It is an enhancement to 6.3.2.1 through 6.3.2.3 and to 6.3.2.5.

Reference:

IEC 61131-3.

6.3.2.5 Code Inspection and Walk Through (CIWT)

Application:

Any feature of the software may be validated with this method if enhanced examination intensity is necessary.

Preconditions:

Source code, text editor, tools. Knowledge of programming languages.

Description:

The examiner walks through the source code assignment by assignment, evaluating the respective part of the code to determine whether the requirements are fulfilled and whether the program functions and features are in compliance with the documentation.

The examiner may also concentrate on algorithms or functions that he has identified as complex, error-prone, insufficiently documented, etc. and inspect the respective part of the source code by analyzing and checking.

Prior to these examination steps the examiner will have identified the legally relevant software part, e.g. by applying the metrological data flow analysis (see 6.3.2.4). In general code inspection or walk through is limited to this part.

Result:

Implementation compatible with the software documentation and in compliance with the requirements or not.

Complementary procedures:

This is an enhanced method, additional to 6.3.2.1 and 6.3.2.4. Normally it is only applied in spot checks.

Reference:

IEC 61508-52010 [9].

6.3.2.6 Software Module Testing (SMT)

Application:

Only if a high conformity and protection against fraud is required. This method is applied when functions of a program cannot be examined exclusively on the basis of written information. It is appropriate and effective in validation of dynamic measurement algorithms.

Preconditions:

Source code, development tools, functioning environment of the software module under test, input dataset and corresponding nominal output dataset or tools for automation. Skills in IT, knowledge of programming languages. Co-operation with the programmer of the module under test is advisable.

Description:

The software module under test is integrated in a test environment, i.e. a specific test program that calls the module under test and provides it with all necessary input data. The test program receives actual output data from the module under test and compares them with the nominal values.

Result:

Module under test is correct or not.

Complementary procedures:

This is an enhanced method, additional to 6.3.2.2 or 6.3.2.5. It is only effective in exceptional cases.

Reference:

IEC 61508-52010 [9].

## Validation procedure

The validation procedure consists of a combination of analysis methods and tests. The relevant OIML Recommendation may specify details concerning the validation procedure, including:

(a) which of the validation methods described in 6.3 shall be carried out for the requirement under consideration;

(b) how the evaluation of test results shall be performed;

(c) which result should be included in the test report and which should be integrated in the test certificate (see Annex B).

In Table 2 two alternative examination levels A and B for the validation procedures are defined. DFA, CIWT and SMT methods are only suggested for level B. . Level B implies an extended examination compared to A. A selection between A and B type validation procedures may be made in the relevant OIML Recommendation – different or equal for each requirement – in accordance with the expected:

* risk of fraud;
* area of application;
* required conformity to approved type;
* risk of wrong measurement result due to operating errors.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Requirement** | | **Validation procedure A**  (normal examination  level) | **Validation procedure B**  (extended examination level) | **Comment** |
| 5.1.1 | Software identification | AD + VFTSw | AD + VFTSw + CIWT | Select “B” if high conformity is required |
| 5.1.2 | Correctness of algorithms and functions | AD + VFTM | AD + VFTM + CIWT/SMT |  |
| **Software protection** | |  |  |  |
| 5.1.3.1 | Prevention of misuse | AD + VFTSw | AD + VFTSw |  |
| 5.1.3.2 | Fraud protection | AD + VFTSw | AD + VFTSw + DFA/CIWT/SMT | Select “B” in case of high risk of fraud |
| **Support of hardware features** | | | | |
| 5.1.4.1 | Support of fault detection | AD + VFTSw | AD + VFTSw + CIWT + SMT | Select “B” if high reliability is required |
| 5.1.4.2 | Support of durability protection | AD + VFTSw | AD + VFTSw + CIWT + SMT | Select “B” if high reliability is required |
| **Specifying and separating relevant parts and specifying interfaces of parts** | | | | |
| 5.2.1.1 | Separation of componentscomponents | AD | AD + DFA/CIWT |  |
| 5.2.1.2 | Separation of software parts | AD | AD + DFA/CIWT |  |
| 5.2.2 | Shared indications | AD + VFTM/ VFTSw | AD + VFTM/ VFTSw + DFA/CIWT |  |
| 5.2.3 | Storage of data | AD + VFTSw | AD + VFTSw + CIWT/SMT | Select “B” if storage of measurement data in unsecure storages is foreseen |
| 5.2.3.1 | The measurement value stored shall be accompanied by all relevant information necessary for future legally relevant use | AD + VFTSw | AD + VFTSw + CIWT/SMT | Select “B” in case of high risk of fraud |
| 5.2.3.2 | The data shall be protected by software means to guarantee authenticity, integrity and, if necessary, correctness of the information of the time of measurement | AD + VFTSw | AD + VFTSw + SMT |  |
|  |  |  |  |  |
| 5.2.3.3 | Automatic storing | AD + VFTSw | AD + VFTSw + SMT |  |
|  |  |  |  |  |
| 5.2.3.4 | Time stamp | AD + VFTSw | AD + VFTSw + SMT |  |
| 5.2.4 | Transmission via communication lines | AD + VFTSw | AD + VFTSw + CIWT/SMT | Select “B” if transmission of measurement data in open system is foreseen |
| 5.2.4.1 | The measurement value transmitted shall be accompanied by all relevant information necessary for future legally relevant use. | AD + VFTSw | AD + VFTSw + CIWT/SMT | Select “B” in case of high risk of fraud |
| 5.2.4.2 | The transmitted data shall be protected by software means to guarantee the authenticity, integrity and, if necessary correctness of the information concerning the time of measurment. | AD + VFTSw | AD + VFTSw + SMT / |  |
| 5.2.4.3 | Transmission delay | AD + VFTSw | AD + VFTSw + SMT | Select "B" in case of high risk of fraud, e.g. transmission in open systems |
| 5.2.4.4 | Transmission interruption | AD + VFTSw | AD + VFTSw + SMT | Select "B" in case of high risk of fraud, e.g. transmission in open systems |
| 5.2.4.5 | Time stamp | AD + VFTSw | AD + VFTSw + SMT |  |
| 5.2.5 | Compatibility of operating systems and hardware | AD + VFTSw | AD + VFTSw + SMT |  |
| **Maintenance and re-configuration** | | | | |
| 5.2.7.2 | Verified Update | AD | AD |  |
| 5.2.7.3 | Traced Update | AD + VFTSw | AD + VFTSw + CIWT/SMT | Select “B” in case of high risk of fraud |

Table 2: Recommendations for combinations of analysis and test methods for the various software requirements (acronyms defined in Table 1)

## Equipment under test (EUT)

Normally, tests are carried out on the complete measuring instrument (functional testing). If the size or configuration of the measuring instrument does not lend itself to testing as a whole unit or if only a separate component or software module of the measuring instrument is concerned, the relevant OIML Recommendation may indicate that the tests, or certain tests, shall be carried out on the components or software modules separately, provided that, in the case of tests with the components or software modules in operation, these are included in a simulated setup, sufficiently representative of its normal operation. The applicant is responsible for the provision of all the required equipment and specimens.

# Verification

If metrological control of measuring instruments is prescribed in a country, there shall be means to check in the field during operation the identity of the software, the validity of the adjustment and the conformity to the approved type.

The relevant OIML Recommendation may require carrying out the verification of the software in one or more stages according to the nature of the considered measuring instrument.

The verification of the software shall include:

* an examination of the conformity of the software with the approved version (e.g. check of the software identification, check of securing means);
* an examination that the configuration is compatible with the declared minimal configuration, if given in the evaluation certificate;
* an examination that the inputs/outputs of the measuring instrument are free of unwanted side effects;
* an examination that the device specific parameters (especially the adjustment parameters) are correct.

# Risk assessment

8.1This clause is intended as a guide to determine a set of risk levels to be generally applied for tests carried out on electronic measuring instruments. It is not intended as a classification with strict limits leading to special requirements as in the case of an accuracy classification.

Moreover, this guide does not restrict the project groups from providing risk assessments that differ from those resulting from the guidelines set forth in this Document. Different risk levels may be used in accordance with special limits prescribed in the relevant OIML Recommendations.

8.2When selecting risk levels for a particular category of instruments and area of application (trade, direct selling to the public, health, law enforcement, etc.), the following aspects can be taken into account:

(a) risk of fraud:

* the consequence and the social and societal impact of malfunction;
* the value of the goods to be measured;
* platform used (built for purpose or universal computer);
* exposure to sources of potential fraud (unattended self-service device).

(b) required conformity:

* the practical possibilities for the industry to comply with the prescribed level.

(c) required reliability:

* environmental conditions;
* the consequence and the social and societal impact of errors.

(d) interest of the defrauder:

* simply being able to commit fraud can be a sufficient motivational factor.

(e) the possibility to repeat a measurement or to interrupt it.

Throughout the requirements clauses (see 5) various examples for acceptable technical solutions are given illustrating the basic level of protection against fraud, conformity, reliability, and type of measurement (marked with (I)). Where suitable, examples with enhanced counter measures are also presented that consider a raised risk level of the aspects described above (marked with (II)).

The validation procedure and risk level are linked. A deep analysis of the software shall be performed when a raised risk level is required in order to detect software deficiencies or security weaknesses. On the other hand, mechanical sealing (e.g. sealing of the communication port or the housing) should be considered when choosing the validation procedure.

**Annex A**

**Bibliography**

At the time of publication, the editions indicated were valid. All referred documents are subject to revision, and the users of this Document are encouraged to investigate the possibility of applying the most recent editions of the referred documents indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

The actual status of the Standards referred to can also be found on the Internet:

IEC Publications: <http://www.iec.ch/searchpub/cur_fut.htm>

ISO Publications: <http://www.iso.org>

OIML Publications: <https://www.oiml.org/en/publications/>

(with free download of PDF files).

In order to avoid any misunderstanding, it is highly recommended that all references to Standards in OIML Recommendations and International Documents be followed by the version referred to (generally the year or date).

|  |  |  |
| --- | --- | --- |
| **Ref.** | **Standards and reference documents** | **Description** |
| [1] | OIML V 2-200:2012 International Vocabulary of Metrology - Basic and General Concepts and Associated Terms (VIM),), 3rd Edition | Vocabulary, developed by the Joint Committee for Guides in Metrology (JCGM), |
| [2] | OIML B 3:2011  OIML Basic Certificate System for OIML Type Evaluation of Measuring Instruments | The OIML Certificate System for Measuring Instruments is a system for issuing, registering and using OIML Certificates of Conformity for types of measuring instruments based on the requirements of OIML Recommendations. |
| [3] | OIML D 11:2013  General requirements for measuring instruments – Environmental conditions | Guidance for establishing appropriate metrological performance testing requirements for influence quantities that may affect the measuring instruments covered by OIML Recommendations. (EMC, climatic, mechanical influences) |
| [4] | ISO/IEC 9594-8:2014  Information technology -- Open Systems Interconnection -- The Directory: Part 8: Public- key and attribute certificate frameworks | ISO/IEC 9594-8:2014 specifies frameworks and a number of data objects that can be used to authenticate and secure the communication between two entities, e.g. between two directory service entities or between a web browser and a web server. The data objects can also be used to prove the source and integrity of data structures such as digitally signed documents. |
| [5] | ISO 2382-9:1995  Information technology -- Vocabulary -- Part 9: Data communication | Intended to facilitate international communication in data communication. Presents terms and definitions of selected concepts relevant to the field of data communication and identifies relationships among the entries. |
| [6] | IEC 61508-4:2010  Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4:  Definitions and abbreviations | Contains the definitions and explanation of terms that are used in parts 1 to 7 of this Standard. Intended for use by Technical Committees in the preparation of Standards in accordance with the principles contained in IEC Guide 104 and ISO/IEC Guide 51. IEC 61508 is also intended as a stand-alone Standard. |
| [7] | ISO/IEC 14598 series  Information technology -- Software product evaluation | The ISO/IEC 14598 series of Standards gives methods for measurement, assessment and evaluation of software product quality. They describe neither methods for evaluating software production processes nor methods for cost prediction (software product quality measurements may, of course, be used for both these purposes). |
| [8] | OIML V 1:2013  International vocabulary of terms in legal metrology (VIML) | The VIML includes only the concepts used in the field of legal metrology. These concepts concern the activities of the legal metrology service, the relevant documents, as well as other problems linked with this activity. Also included in this Vocabulary are certain concepts of a general character which have been drawn from the VIM. |
| [9] | IEC 61508-5:20105  Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 5: Examples of methods for the determination of safety integrity levels | Provides information on the underlying concepts of risk and the relationship of risk to safety integrity (see Annex A); a number of methods that will enable the safety integrity levels for the E/E/PE safety-related systems, other technology safety-related systems and external risk reduction facilities to be determined (see Annexes, B, C, D and E). Intended for use by Technical Committees in the preparation of Standards in accordance with the principles contained in IEC Guide 104 and ISO/IEC Guide 51. |
| [100] | WELMEC Guide 2.3, May 2005 Issue 3  Guide for Examining Software (Weighing Instruments) |  |
| [111] | WELMEC Guide 7.2, Issue 2015  Software Guide (Measuring Instruments Directive 2014/32/EU) | This document provides guidance to all those concerned with the application of the Measuring Instruments Directive (European Directive 2014/32/EU; MID), especially for software-equipped measuring instruments. It addresses both manufacturers of measuring instruments and notified bodies which are responsible for conformity assessment of MID instruments. By following the Guide, compliance with the software- related requirements contained in the MID can be assumed. |

**Annex B**

**Example of a software evaluation report**

**(Informative)**

*Note:* The Technical Committees and Subcommittees developing OIML Recommendations should decide which information shall be included in Test Report and OIML Certificate of Conformity. E.g. the name, version and checksum of the executable file from the following example should be included in the Test Certificate.

**Test report no** XYZ122344

**Validation of Software of the flow meter** Tournesol Metering **model** TT100

The software of the measuring instrument was validated to show conformance with the requirements of the OIML Recommendation R-xyz.

The validation was based on the report OIML International Document D 31:YYYY, where the essential requirements for software are interpreted and explained. This report describes the examination of software needed to state conformance with the R-xyz.

Manufacturer Applicant

Tournesol Metering New Company

P.O. Box 1120333 Nova Street 123

100 Klow 1000 Las Dopicos

Syldavie San Theodorod

Reference: Mr. Tryphon Tournesol Reference: Archibald Haddock

**Test object**

The Tournesol Metering meter TT100 is a measuring instrument intended to measure flow in liquids. The intended range is from 1 L/s up to 2000 L/s. The basic functions of the instrument are:

* measuring of flow in liquids,
* indication of measured volume,
* interface to transducer.

The flow meter is described as a built-for-purpose measuring instrument (an embedded system) with a storage device containing legally relevant data.

The flow meter TT100 is an independent instrument with a transducer connected. The transducer incorporates a temperature compensation. Adjustment of flow rates is possible by calibration parameters stored in a non-volatile memory of the transducer. It is fixed to the instrument and cannot be disconnected. The measured volume is indicated on a display. No communication with other devices is possible.

The embedded software of the measuring instrument was developed by

**Tournesol Metering, P.O. Box 1120333, 100 Klow, Syldavie.**

The executable file name is “**tt100\_12.exe**”.

The validated version of this software is **V1.2c**. The software version is presented on the display upon instrument start-up and by pressing the “level” button for 4 seconds.

The source code comprises the following legally relevant files:

* main.c 12301 byte 23 Nov 2003;
* int.c 6509 byte 23 Nov 2003;
* filter.c 10897 byte 20 Oct 2003;
* input.c 2004 byte 20 Oct 2003;
* display.c 32000 byte 23 Nov 2003;
* ethernet.c 23455 byte 15 June 2002;
* driver.c 11670 byte 15 June 2002;
* calculate.c 6788 byte 23 Nov 2003.

The executable file “**tt100\_12.exe**” is protected against modification by a checksum. The value of the checksum by algorithm **XYZ** is **1A2B3C**.

The validation was supported by the following documents from the manufacturer:

* TT 100 User Manual Release 1.6;
* TT 100 Maintenance Manual Release 1.1;
* Software description TT100 (internal design document, dated 22 Nov 2003);
* Electronic circuit diagram TT100 (drawing no 222-31, dated 15 Oct 2003).

The final version of the test object was delivered to the National Testing & Measurement Laboratory on 25 November 2003.

**Results of validation**

The validation was performed according to the OIML D 31:YYYY. The validation was performed between 1 November and 23 December 2003. A design review was held on 3 December by Dr. K. Fehler at Tournesol Metering head office in Klow. Other validation work was carried out at the National Testing & Measurement Laboratory by Dr. K. Fehler and Mr. S. Problème.

**The following requirements were validated:**

* software identification;
* correctness of algorithms and functions;
* software protection;
* prevention against accidental misuse;
* fraud protection;
* support of hardware features;
* storage of data, transmission via communication systems.

**The following validation methods were applied:**

* analysis of the documentation and validation of the design;
* validation by functional testing of metrological features;
* walkthrough, code inspection;
* software module testing of module calculate.c with SDK XXX.

**Result**

The following requirements of the OIML D 31:YYYY were validated without any faults being found:

5.1.1, 5.1.2, 5.1.3.2, 5.2.1, 5.2.2.1, 5.2.2.2, 5.2.2.3.

The result applies to the tested item with Serial No. 1188093-B-2004 only.

**Conclusion**

The software of the **Tournesol Metering TT100 V1.2c** fulfils the requirements of OIML R-xyz.

National Testing & Measurement Lab.

Software Department

Dr. K.E.I.N. Fehler Mr. S.A.N.S. Problème

Technical manager Technical Officer

**Checklist**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clause** | **Requirement** | **Passed** | **Failed** | **Remarks** |
| **5.1**  **5.1.1** | **General requirements**  **Software identification**  Legally relevant software shall be clearly identified. |  |  |  |
| **5.1.2** | **Correctness of algorithms and functions**  The measuring algorithms and functions of a measuring instrument shall be appropriate and functionally correct. |  |  |  |
| **5.1.3**  **5.1.3.1** | **Software protection**  **Prevention of misuse**  A measuring instrument shall be constructed in such a way that possibilities for unintentional, accidental, or intentional misuse are minimal. |  |  |  |
| **5.1.3.2**  **a)**  **b)**  **c)**  **d)** | **Fraud protection**  Software shall be protected in such a way that evidence of any intervention (e.g. software updates, parameter changes) shall be available. Software shall be secured against unauthorized modification, loading, or changes by swapping the memory device.  Only clearly documented functions may be activated by the user interface, which do not influence the metrological characteristics of the instrument.  Parameters that fix the legally relevant characteristics of the measuring instrument shall be secured against unauthorized modification. If necessary for the purpose of verification, displaying or printing of the current parameter settings shall be possible.  Software protection comprises appropriate sealing by mechanical, electronic and/or cryptographic means, making an unauthorized intervention impossible or evident. |  |  |  |
| **5.1.4**  **5.1.4.1** | **Support of hardware features**  **Support of fault detection**  The manufacturer of the instrument shall be required to design checking facilities into the software or hardware parts or provide means by which the hardware parts can be supported by the software parts of the instrument. |  |  |  |
| **5.1.4.2** | **Support of durability protection**  It is the manufacturer’s choice to realize durability protection facilities in software or hardware, or to allow hardware facilities to be supported by software. |  |  |  |
| **5.2**  **5.2.1** | **Requirements specific for configurations**  **Specification and separation of legally relevant parts and specification of interfaces**  Legally relevant parts of a measuring instrument shall not be inadmissibly influenced by other parts of the measuring instrument. |  |  |  |
| **5.2.1.1**  **a)**  **b)** | **Separation of components**  Components of a measuring instrument that perform legally relevant functions shall be identified, clearly defined, and documented.  It shall be demonstrated that the functions and data of legally relevant components cannot be inadmissibly influenced by commands received via the interface to the other, legally non-relevant parts. |  |  |  |
| **5.2.1.2**  **a)**  **b)** | **Specification and separation of software parts**  The conformity requirement applies to the legally relevant software part of a measuring instrument (see 5.2.6) and it shall be made identifiable as described in 5.1.1.  If the legally relevant software part communicates with other software parts, a software interface shall be defined. All communication shall be performed exclusively via this interface. The legally relevant software part and the interface shall be clearly documented. All legally relevant functions and data domains of the software shall be described to enable a type evaluation authority to decide on correct software separation. |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clause** | **Requirement** | **Passed** | **Failed** | **Remarks** |
| **c)**  **d)** | There shall be an unambiguous assignment of each command to all initiated functions or data  changes in the legally relevant part of the software. Commands that communicate through the software interface shall be declared and documented. Only documented commands are allowed to be activated through the software interface.  Where legally relevant software has been separated from non-relevant software, the legally relevant software shall have priority using the resources over non-relevant software. |  |  |  |
| **5.2.2** | **Shared indications**  If a display or printout is used both for legally relevant and legally non-relevant outputs, the legally relevant information should always be readable, and clearly distinguishable from other informaton. |  |  |  |
| **5.2.3**  **5.2.3.1** | **Storage of data**  The measurement value stored shall be accompanied by all relevant information necessary for future legally relevant use. |  |  |  |
| **5.2.3.2** | The stored data shall be protected by software means to guarantee authenticity, integrity and, if necessary correctness of the information concerning the time of measurement. |  |  |  |
| **5.2.3.3**  **a)**  **b)** | **Automatic storing**  When data storage is required, measurement data must be stored automatically when the measurement is concluded. The storage device must have sufficient permanency to ensure that the data are not corrupted under normal storage conditions. There shall be sufficient memory storage for any particular application.  Stored data may be deleted if either:  - the transaction is settled;  - these data are printed by a printing device subject to legal control. |  |  |  |
| **5.2.3.4** | **Time stamp**  The time stamp shall be read from the clock of the instrument. Appropriate protection means shall be taken according to the risk level to be applied. If the information concerning the time of measurement is necessary, the reliability of the internal clock of the measuring instrument shall be enhanced by specific means. |  |  |  |
| **5.2.4** | **Transmission via communication lines** |  |  |  |
| **5.2.4.1** | The measurement value transmitted shall be accompanied by all relevant information necessary for future legally relevant use. |  |  |  |
| **5.2.4.2** | The transmitted data shall be protected by software means to guarantee the authenticity, integrity and, if necessary correctness of the information concerning the time of measurement. |  |  |  |
| **5.2.4.3** | The measurement shall not be inadmissibly influenced by a transmission delay. |  |  |  |
| **5.2.4.4** | If network services become unavailable, no measurement data shall be lost. |  |  |  |
| **5.2.4.5** | The time stamp shall be read from the clock of the instrument. Appropriate protection means shall be taken according to the risk level to be applied. If the information concerning the time of measurement is necessary, the reliability of the internal clock of the measuring instrument shall be enhanced by specific means. |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Clause** | **Requirement** | **Passed** | **Failed** | **Remarks** |
| **5.2.5**  **5.2.5.1** | **Compatibility of operating systems and hardware**  The manufacturer shall state the legally relevant components of the operating system. The configuration of the operating system shall be identifiable. Changes to the configuration of the operating system shall be traceable. |  |  |  |
| **5.2.5.2** | The manufacturer shall identify the hardware and software environment that is suitable. Minimum resources and a suitable configuration necessary for correct functioning shall be declared by the manufacturer. |  |  |  |
| **5.2.5.3** | Technical means shall be provided to prevent operation, if the minimal configuration requirements are not met. |  |  |  |
| **5.2.7**  **5.2.7.1** | **Maintenance and reconfiguration**  Only versions of legally relevant software that conform to the approved type are allowed for use. |  |  |  |
| **5.2.7.2** | **Verified Update**  After the update of the legally relevant software of a measuring instrument (exchange with another approved version or re-installation) the measuring instrument is not allowed to be employed for legal purposes before a verification of the instrument has been performed and the securing means have been renewed. |  |  |  |
| **5.2.7.3**  **a)**  **b)**  **c)**  **d) e) f)**  **g)**  **h)** | **Traced Update**  Traced Update of software shall be automatic. If some of the securing measures of the instrument are turned off to enable updating, they must be turned on again immediately after update, independent of the result of the update process.  Software shall be protected in such a way that evidence of any intervention shall be available. During an update, any existing audit trail information shall be retained.  Technical means shall be employed to guarantee the authenticity of the loaded software.  Technical means shall be employed to ensure the integrity of the loaded software, i.e. that it has not been inadmissibly changed before loading.  Appropriate technical means shall be employed to ensure that Traced Updates are adequately traceable within the instrument.  Depending on national legal legislation it may be necessary for the user or owner of the measuring instrument to have to give his consent.  If the requirements 5.2.7.3.a through 5.2.7.3.f cannot be fulfilled, it is still possible to update the legally non-relevant software part. In this case the following requirements shall be met:  - there is a distinct separation between the legally relevant and non-relevant software according to 5.2.1;  - the whole legally relevant software part cannot be updated without breaking a seal;  - it is stated in the type evaluation certificate that updating of the legally non-relevant part is acceptable.  If the loaded software fails integrity test or authenticity test, the instrument shall discard the new version and use the previous version of the software or switch to an inoperable mode. |  |  |  |
| **5.2.7.4** | The measuring instrument shall be fitted with a facility to automatically and non-erasably record any adjustment of the device specific parameter, e.g. an audit trail. The instrument shall be capable of presenting the recorded data. |  |  |  |
| **5.2.7.5** | The audit trails are part of the legally relevant software and should be protected as such. |  |  |  |

**Annex C**

**Index**

**Acceptable solution:** 5.1; 5.1.1; 5.2; 5.2.1.2.d; 8.2.

**Audit trail:** 3.1.1; 5.1.3.2.d; 5.2.7.3; 5.2.7.3.b; 5.2.7.3.e; 5.2.7.3.h; 5.2.7.4; 5.2.7.5; 6.1.1; 6.2.1.

**Authentication:** 3.1.2.; 3.1.3; 5.2.7.3.

**Authenticity:** 3.1.3; 3.1.12; 5.1.3.2.d; 5.2.3.2; 5.2.4.2; 5.2.7.3.c.; 5.2.7.3.h.

**Checking facility:** 3.1.4; 5.1.4.1.

**Commands:** 3.1.36; 5.1.3.2.b; 5.2.1.1.b; 5.2.1.2.b; 5.2.1.2.c; 6.1; 6.1.1; 6.2; 6.3.1; 6.3.2.1; 6.3.2.3; 6.3.2.4; Annex B.

**Communication:** 3.1.5; 3.1.43; 5.2.1.2.b; 5.2.1.2.c; 5.2.1.2.d; 5.2.4; 5.2.5.2; 6.3.1; 6.3.2.1; 6.4; 8.2; Annex B.

**Communication interface:** 3.1.5; 5.1.1.

**Cryptographic certificate:** 3.1.6; 3.1.12; 5.1.3.2.d.

**Cryptographic means:** 3.1.7; 5.1.3.2.d; 5.2.7.3.c; Annex B.

**Data domain:** 3.1.8; 3.1.36; 3.1.37; 3.1.38; 5.2.1.2.b; 5.2.3.3.a; 6.3.2.4; Annex B.

**Device-specific parameter:** 3.1.9; 3.1.25; 5.1.3.2.c; 5.2.7.1; 5.2.7.4.

**Durability:** 3.1.10; 5.1.4.2; 6.1.1; 6.4; Annex B.

**Electronic measuring instrument:** 3.1.11; 3.1.18; 8.1.

**Electronic device:** 3.1.25; 5.2.1.1.a; 5.2.1.2.d.

**Electronic Signature:** 3.1.7; 3.1.12; 5.1.3.2.d; 5.2.3.2; 5.2.4.2; 5.2.7.3.c.

**Error (of indication):** 3.1.13; 3.1.18; 3.1.23; 6.1.1; 6.2; 6.3.1; 6.3.2.5; 6.4; 8.2.

**Error log:** 3.1.14; 5.1.4.1.

**Evaluation:** 3.1.2; 3.1.44; 3.1.45; 3.2; 5.1.1; 5.1.3.2.c; 5.1.4.1; 5.2.1.1.a; 5.2.1.2.b; 5.2.5.2; 5.2.7.3.c; 5.2.7.3.g; 6.1; 6.1.1; 6.2; 6.3.1; 6.3.2.1; 6.4; 7; Annex B.

**Event:** 3.1.1; 3.1.15; 3.1.16; 3.1.33; 3.1.42; 5.1.3.2.d; 5.2.1.2.d; 5.2.7.3.e; 6.2.1.

**Event counter:** 3.1.16; 5.1.3.2.d; 6.2.1.

**Executable code:** 3.1.17; 3.1.40.; Annex B

**Fault:** 3.1.14; 3.1.18; 3.1.33; 3.1.42; 5.1.4.1; 6.1.1; 6.3.1 ; 6.3.2.1 ; 6.3.2.3 ; 6.4 ; Annex B.

**Hash function:** 3.1.7; 3.1.19; 5.1.1; 5.1.4.1; 5.2.33; 5.2.7.3.d.

**Integrity of programs, data, or parameters:** 3.1.12; 3.1.20; 5.2.3.2; 5.2.4.2; 5.2.7.3; 5.2.7.3.d; 5.2.7.3.h; 6.4; Annex B.

**Interface:** 3.1.5; 3.1.21; 5.1.1; 5.2.1; 5.2.1.1.a; 5.2.1.1.b; 5.2.1.2.b; 5.2.1.2.c; 5.2.1.2.d; 5.2.5.1.a; 5.2.5.1.b; 6.1.1; 6.3.2.1; 6.3.2.3; 6.3.2.4; 6.4; Annex B.

**Intrinsic error:** 3.1.18; 3.1.23.

**Legally relevant:** 3.1.1; 3.1.24; 3.1.25; 3.1.26; 3.1.36; 5.1.3.1; 5.1.3.2.a; 5.1.3.2.c; 5.1.4.1; 5.2.1.1.a; 5.2.1.1.b; 5.2.1.2.a; 5.2.1.2.b; 5.2.1.2.c; 5.2.1.2.d; 5.2.2; 5.2.3.1; 5.2.3.4; 5.2.4.1; 5.2.4.5; 5.2.5.1.a; 5.2.5.3; 5.2.7; 6.1.1; 6.4; Annex B.

**Legally relevant parameter:** 3.1.25; 3.1.45; 5.1.3.2.d; 5.1.4.1.

**Legally relevant software part:** 3.1.26; 3.1.39; 3.1.45; 5.1.3.2.a; 5.1.3.2.b; 5.1.4.1; 5.2.1; 5.2.1.1.a; 5.2.1.2.a; 5.2.1.2.b; 5.2.1.2.c; 5.2.1.2.d; 5.2.2; 5.2.3.2; 5.2.4.2; 5.2.5.3; 5.2.6; 5.2.7; 5.2.7.1; 5.2.7.2; 5.2.7.3.e; 5.2.7.3.g; 5.2.7.5; 6.1; 6.1.1; 6.3.2.5; Annex B.

**Maximum permissible error:** 3.1.27; 3.2.

**Measuring instrument:** 1; 2.1; 2.2; 2.3; 3.1.1; 3.1.2; 3.1.4; 3.1.5; 3.1.6; 3.1.10; 3.1.11; 3.1.15; 3.1.17; 3.1.18; 3.1.25; 3.1.26; 3.1.27; 3.1.28; 3.1.31; 3.1.33; 3.1.37; 3.1.38; 3.1.39; 3.1.44; 3.1.45; 3.1.47; 4.3; 5.1; 5.1.1; 5.1.2; 5.1.3.1; 5.1.3.2.a; 5.1.3.2.c; 5.1.3.2.d; 5.1.4.2; 5.2; 5.2.1; 5.2.1.1.a; 5.2.1.2.a; 5.2.2; 5.2.3.1; 5.2.3.2; 5.2.3.4; 5.2.4.2; 5.2.4.5; 5.2.5.1; 5.2.5.1.b; 5.2.7; 5.2.7.2; 5.2.7.3; 5.2.7.3.c; 5.2.7.3.f; 5.2.7.4; 6.1; 6.1.1; 6.3.2.1; 6.3.2.2; 6.5; 7; 8.1; Annex B.

**Non-interruptible/interruptible measurement:** 3.1.22; 3.1.29; 5.1.4.1. 5.2.5.1; 5.2.5.1.a; 5.2.5.1.b; 5.2.5.1.c

**Operating system:** 5.1.3.2.a; 5.2.1.1.b; 5.2.1.2.c; 5.2.1.2.d; 5.2.5.2;

**Performance:** 3.1.10; 3.1.30; 6.2;

**Program code:** 3.1.36; 3.1.40; 3.1.43; 5.1.4.1; 5.2.1.2.b; 5.2.3.2; 5.2.4.2.

**Sealing:** 3.1.31; 5.1.3.2.a; 5.1.3.2.d; 6.1.1; 8.2; Annex B.

**Securing:** 3.1.12; 3.1.32; 5.2.1.1.a; 5.2.1.1.b; 5.2.2; 5.2.7.2; 5.2.7.3.a; 6.2.1; 7; Annex B.

**Software examination:** 3.1.34; 5.1.2; 6.3.

**Software identification:** 3.1.35; 5.1.1; 5.2.7.3.e; 6.1.1; 6.2.1; 6.3.2.3; 6.4; 7; Annex B.

**Software interface:** 3.1.36; 3.1.39; 5.2.1.2.b; 5.2.1.2.c; 6.1; 6.1.1; 6.3.2.4; Annex B.

**Software module:** 3.1.8; 3.1.15; 3.1.25; 3.1.26; 3.1.36; 3.1.37; 5.2.1.2.a; 5.2.3.2; 5.2.4.2; 6.1.1; 6.3.1; 6.3.2.6; 6.5; Annex B.

**Software protection:** 3.1.18; 5.1.3; 5.1.3.2.d; 6.4; Annex B.

**Software separation:** 3.1.39; 5.2.1.2.b; 5.2.1.2.d; 6.3.1; 6.3.2.4; Annex B.

**Source code:** 3.1.40; 6.1.1; 6.3.1; 6.3.2.2; 6.3.2.4; 6.3.2.5; 6.3.2.6; Annex B.

**Storage device:** 3.1.41; 5.2.3.3.a; 5.2.7.3.e; Annex B.

**Sub-assembly:** 3.1.25;

**Test:** 3.1.30; 5.1.2; 5.2.7.3.h; 6.2; 6.3.1; 6.3.2.1; 6.3.2.2; 6.3.2.3; 6.3.2.6; 6.4; 6.5; 8.1; Annex B.

**Time Stamp:** 3.1.1; 3.1.42; 5.2.1.1.b; 5.2.3.1; 5.2.3.4; 5.2.4.1; 5.2.4.5; 5.2.7.3.e; 6.4; Annex B.

**Transmission of measurement data**: 3.1.43; 5.2.1.1.a; 5.2.4; 5.2.4.2; 5.2.4.3; 5.2.4.4; 6.4; Annex B.

**Type-specific parameter:** 3.1.25; 3.1.45; 5.1.3.2.c.

**Universal computer:** 3.1.46; 5.1.3.2.a; 5.2.1.1.a; 5.2.1.2.c; 5.2.1.2.d; 5.2.2; 5.2.5.3; 8.2.

**User interface:** 3.1.47; 5.1.1; 5.1.3.2.b; 5.2.2; 6.1.1; 6.3.2.3; Annex B.

**Validation:** 3.1.48; 6.1.1; 6.2; 6.3; 6.3.1; 6.3.2; 6.3.2.1; 6.3.2.2; 6.3.2.3; 6.3.2.6; 6.4; 8.2; Annex B.

**Verification:** 3.1.48; 3.1.49; 5.1.3.2.c; 5.2.7; 5.2.7.1; 5.2.7.2; 5.2.7.3; 5.2.7.3.e; 5.2.7.3.h; 6.2; 7; Annex B.