OIML TC5/SC2/N7
2006-01-27
Replaces N4

Document OIML D-SW

Working Draft

1 WD

General Requirements for Software Controlled Measuring Instruments

Exigences générales pour les instruments de mesure contrôlés par logiciel

Original Version in: English

OIML TC5/SC2 Secretariat: Germany and France

Collaborating members: Slovenia

Participants: ...

Observing: ...

Liaison: ...

Explanatory Note

Reasons for edition of this document

- Overall need for preparation of a document concerning software of measuring instruments, declared as high priority OIML project.
- Existing OIML R-documents have been checked. It turned out there were few concerning software explicitly and quite some requirements concerning software implicitly but they were unbalanced (between different recommendations)
- There are lot of international standards concerning IT, but they do not fit to legal metrology issues completely.
- Referring to standards (quotation of standards: i.e. loading and updating of software)
- Process related approach is perhaps not usual for other metrological fields.

History of development, changes in the course of the development:

There were three major inputs during preparation of the first pre-draft:

- Answers of the SC2 members to the questionnaire
  - After analysis of the results of the Query the importance of the issues was ranked (see annual report 2003)
  - table with ranking (see Annex E)
  - note: no member opposed to this approach
- Analysis of software related requirement in OIML R documents
- Present experience of member countries
  - List of important ideas of member countries introduced into this document
  - Table with an evaluation of the member's comments
  - Existing drafts of regional software requirements for measuring instruments

All of the above mentioned inputs were collected and put into the structure according to OIML instructions on structuring the documents. A cross reference between questionnaire and draft requirements of this Document is given in Annex E. Cross references to the draft Canadian Specification for Metrological Software and to the European software requirements based on the Measuring Instruments Directive MID are given in Annexes F and G respectively.

Contents

FOREWORD

The International Organisation of Legal Metrology (OIML) is a worldwide, intergovernmental organisation whose primary aim is to harmonise the regulations and metrological controls applied by the national metrological services, or related organisations, of its Member States.

The two 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; the OIML Member States shall implement these Recommendations to the greatest possible extent;
- International Documents (OIML D), which are informative in nature and intended to improve the work of the metrological services.

OIML Draft Recommendations and Documents are developed by technical committees or subcommittees which are formed by the Member States. Certain international and regional institutions also participate on a consultation basis.

Cooperative agreements are established between OIML and certain institutions, such as ISO and IEC, with the objective of avoiding contradictory requirements; consequently, manufacturers and users of measuring instruments, test laboratories, etc. may apply simultaneously OIML publications and those of other institutions.

International Recommendations and International Documents are published in French (F) and English (E) and are subject to periodic revision.

This publication – pre-draft D-SW, edition 1 (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 xxxx.*~~

OIML publications may be obtained from the Organisation's headquarters:

Bureau International de Métrologie Légale
11, rue Turgot - 75009 Paris - France
Telephone: 33 (0)1 48 78 12 82 and 42 85 27 11
Fax: 33 (0)1 42 82 17 27
E-mail: biml@oiml.org
Internet: www.oiml.org

1 Introduction

The primary aim of this International Document is to provide the OIML technical committees and subcommittees with guidance for establishing appropriate requirements for software-related functionality 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.

2 Scope and field of application

2.1 This International Document specifies the general requirements applicable to software related functionality 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 particular software requirements and procedures to be specified in International Recommendations applicable to particular categories of measuring instruments (hereafter in brief: relevant Recommendation).

2.3 The instructions given in this document apply only to software-controlled measuring instruments or electronic devices.

Notes:

(1) This Document does not cover all of the technical requirements specific for that kind of measuring instrument; these requirements are to be given in the relevant Recommendation, eg. for weighing instruments, water meters, ...

(2) This Document addresses some aspects concerning data security. In addition, national regulations for this areas have to be considered.

(3) As software controlled devices are always electronic, it is necessary to consider OIML D11 (“General requirements for electronic measuring instruments”) as well.

3 Terminology

Some of the definitions used in this International Document are in conformity with the International vocabulary of basic and general terms in metrology (VIM) [1]. For the purpose of this International Document, the following definitions and abbreviations apply.

3.1 General terminology

3.1.1 Electronic measuring instrument [D 11, 3.1]

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

Note:

For the purpose of this Document, auxiliary equipment, as far as subject to metrological control, is considered to be part of the measuring instrument.

3.1.2 Electronic device [D 11, 3.2]

A device employing electronic sub-assemblies and performing a specific function. Electronic devices are usually manufactured as a separate unit and are capable of being tested independently.

Notes:

(1) An electronic device may be a complete measuring instrument (for example: counter scale, electricity meter) or a part of a measuring instrument (for example: printer, indicator).

(2) An electronic device can be a module in the sense this term is used in the OIML Publication B3 “The OIML Certificate system for Measuring Instruments” [2].

3.1.3 Electronic sub-assembly [D 11, 3.3]

A part of an electronic device, employing electronic components and having a recognisable function of its own.

Examples: amplifiers, comparators, power converters, storage devices, calculator (NAWI, fuel dispensor, self-service device, checking facilities).

3.1.4 Electronic component [D 11, 3.4]

The smallest physical entity that uses electron or hole conduction in semi-conductors, gases or in a vacuum.

Examples: electronic tubes, transistors, integrated circuits.

3.1.5 Error (of indication) [VIM 5.20, D11, 3.5]

Indication of a measuring instrument minus a true value of the corresponding input quantity. [1])

---

[1]) VIM is being revised. In the present draft, “*error of indication*” is defined in A6

3.1.6 Maximum permissible errors (of a measuring instrument) [VIM 5.21, D11, 3.6]

Extreme values of an error permitted by specifications, regulations, etc. for a given measuring instrument.

3.1.7 Intrinsic error [VIM 5.24, D11, 3.7]

The error of a measuring instrument, determined under reference conditions.[2])

3.1.8 Initial intrinsic error [D 11, 3.8]

The intrinsic error of a measuring instrument as determined prior to performance tests and durability evaluations.

3.1.9 Fault [D 11, 3.9]

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

Notes:

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

(2) From the definition it follows that in this Document, a "fault" is a numerical value which is expressed either in a unit of measurement or as a relative value, for instance in %.

3.1.10 Significant fault [D 11, 3.10]

A fault greater than the value specified in the relevant Recommendation (see 2.2)

Note:

The relevant Recommendation may specify that the following faults are not significant, even when they exceed the value defined in this definition:

(a) faults arising from simultaneous and mutually independent causes (e.g. EM fields and discharges) originating in the measuring instrument or in its checking facilities,

(b) faults implying the impossibility to perform any measurement,

(c) transitory faults being momentary variations in the indication, which cannot be interpreted, memorised or transmitted as a measurement result,

(d) faults giving rise to variations in the measurement result that are serious enough to be noticed by all those interested in the measurement result; the relevant Recommendation may specify the nature of these variations.

3.1.11 Durability error [D 11, 3.11]

The difference between the intrinsic error after a period of use and the initial intrinsic error of a measuring instrument.

---

[2]) VIM is being revised. In the present draft, *"intrinsic error"* is defined in A13.

3.1.12 Significant durability error [D 11, 3.12]

A durability error greater than the value specified in the relevant Recommendation.

Note:

The relevant Recommendation may specify that durability errors are not significant, even when they exceed the value defined in this definition, in the following cases:

(a) the indication cannot be interpreted, memorised or transmitted as a measurement result,

(b) the indication implies the impossibility to perform any measurement,

(c) the indication is so obviously wrong that it is bound to be noticed by all those interested in the result of the measurement, or

(d) a durability error cannot be detected and acted upon due to a breakdown of the appropriate durability protection facility.

3.1.13 Influence quantity [VIM 2.7]

A quantity that is not the measurand but that affects the result of the measurement.

3.1.14 Influence factor [D 11, 3.13.1]

An influence quantity having a value within the rated operating conditions of the measuring instrument specified in the relevant Recommendation.

3.1.15 Disturbance [D 11, 3.13.2]

An influence quantity having a value within the limits specified in the relevant Recommendation, but outside the specified rated operating conditions of the measuring instrument.

Note:

An influence quantity is a disturbance if the rated operating conditions for that influence quantity are not specified.

3.1.16 Rated operating conditions [Adapted from VIM 5.5]

Conditions of use giving the range of values of influence quantities for which specified metrological characteristics of a measuring instrument are intended to lie within given limits.[3])

3.1.17 Reference conditions [VIM 5.7]

Conditions of use prescribed for testing the performance of a measuring instrument or for intercomparison of results of measurements.

Note:

The reference conditions generally include reference values or reference ranges for the influence quantities affecting the measuring instrument.[4])

---

[3]) The VIM is being revised. In the present draft, there is another definition (4.8).

[4]) The VIM is being revised. In the present draft, definition 4.10, *"testing"* has been replaced by "*evaluating*" and there are new notes.

3.1.18 Performance [D 11, 3.16]

The ability of the measuring instrument to accomplish its intended functions.

3.1.19 Durability [D 11, 3.17]

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

3.1.20 Checking facility [D 11, 3.18]

A facility that is incorporated in a measuring instrument and which enables significant faults 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.).

3.1.21 Automatic checking facility [D 11, 3.18.1]

A checking facility that operates without the intervention of an operator.

3.1.22 Permanent automatic checking facility (type P) [D 11, 3.18.1.1]

An automatic checking facility that operates at each measurement cycle.

3.1.23 Intermittent automatic checking facility (type I) [D 11, 3.18.1.2]

An automatic checking facility that operates at certain time intervals or per fixed number of measurement cycles.

3.1.24 Non-automatic checking facility (type N) [D 11, 3.18.2]

A checking facility that requires the intervention of an operator.

3.1.25 Software controlled checking facility

A checking facility that is operated by software (type P, I or N).

3.1.26 Durability protection facility [D 11, 3.19]

A facility that is incorporated in a measuring instrument and which enables significant durability errors to be detected and acted upon.

3.1.27 Test [D 11, 3.20]

A series of operations intended to verify the compliance of the equipment under test (EUT) with specified requirements.

3.1.28 Test procedure [D 11, 3.20.1]

A detailed description of the test operations.

3.1.29 Test program [D 11, 3.20.2]

A description of a series of tests for certain types of equipment.

3.1.30 Performance test [D 11, 3.20.3]

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

3.1.31 Evaluation [VIM]

{Definition to be added according to that in the latest VIM.}

3.1.32 Measuring instrument [VIM, 4.1]

Device intended to be used to make measurements, alone or in conjunction with supplementary device(s).

3.2 Software terminology

3.2.1 Audit trail

A continuous data file containing an information record of the changes to the values of the calibration or configuration parameters of a device, of updates of the software or other activities or events that are legally relevant and may influence metrological characteristics. Every log entry has a unique time and date stamp.

3.2.2 Authentication

Checking of the declared or alleged identity of a user, process, or device.

3.2.3 Authenticity

Result of the process of authentication (passed of failed).

3.2.4 Closed network

A network of a fixed number of participants with a known identity, functionality and location (see also *Open network*).

3.2.5 Commands

Commands may be a sequence of electrical (optical, electromagnetic, etc.) signals on input interfaces or codes in data transmission protocols.

3.2.6 Communication

Exchange of information between two or more units following specific rules.

3.2.7 Communication interface

An electronic, optical, radio or other technical interface that enables information to be automatically passed between components of measuring instruments or subassemblies.

3.2.8 Data domain

It represents parameters, variables, stacks or registers, which are used by programmes to keep values of data. Data domains may belong to one *software module* only or to several.

3.2.9 Device-specific parameter

Legally relevant parameter with a value that depends on the individual instrument. 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).

3.2.10 Executable code

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

3.2.11 Fixed legally relevant software part

Part of the legally relevant software that is and remains identical in the executable code to that of the approved type. [5])

3.2.12 Interface

An interface is a connection part of the device. It allows to establish communication between several devices or sub-assemblies or between several different software modules (see *software interface*).

3.2.13 Integrity of programmes, data, or parameters

Assurance that the programmes, data, or parameters have not been subjected to any unauthorised or unintended changes while in use, transfer, storage, repair or maintenance.

3.2.14 IT configuration

Design of the measuring instrument with respect to IT (Information Technology) functions and features that are – as regards the requirements – independent from the measurement function. The terms are accordingly applicable to sub-assemblies.

3.2.15 Legally relevant parameter

Parameter of a measuring instrument or a sub-assembly subject to legal control. The following types of legally relevant parameters can be distinguished: *type-specific parameters* and *device-specific parameters*.

3.2.16 Legally relevant software part

The part of all *software modules* of a measuring instrument, device, or sub-assembly that defines or fulfils functions or represents features which are subject to legal control (see also *Fixed legally relevant software part*).

---

[5]) This part is eg. responsible to monitor software update (loading software, authentication, integrity checking, installation and activation.

3.2.17 Long-term storage of measurement data

Storage used for keeping measurement data ready after completion of the measurement for later legally relevant purposes (eg. the conclusion of a commercial transaction).

3.2.18 Open network

A network of arbitrary participants (devices with arbitrary functions). The number, identity and location of a participant can be dynamic and unknown to the other participants (see also *Closed network*).

3.2.19 Programme code

Source code or *executable code*.

3.2.20 Software identification

A sequence of readable characters (eg. version number, checksum) that is inextricably linked to the software or *software module* under consideration. It can be checked at an instrument in use.

3.2.21 Software interface

It consists of programme code and a dedicated data domain. It receives, filters, or transmits data between the *legally relevant software part* and other *software modules*.

3.2.22 Software module [similar IEC 61508-4, 3.3.7]

Logic entities (programmes, subroutine, libraries, objects, ...) of subroutines and *data domains* that may be in relationship with other entities. The software of measuring instruments, devices or sub-assemblies consists of one or more software modules.

3.2.23 Software protection

Securing of measuring instrument software or data domain by physical seal or by hardware or software implemented seal. The seal has to be removed, damaged or broken to get access to change software or data domain.

3.2.24 Software separation

Software in measuring devices can be divided into a *legally relevant part* and a legally irrelevant part. These parts communicate via a *software interface*.

3.2.25 Source code

Computer programme written in a form (programming language) which is legible and editable. Source code is compiled or interpreted into *executable code*.

3.2.26 Sub-assembly [D11, 3.3]

See Electronic sub-assembly (3.1.3).

3.2.27 Transmission of measurement data

Transmission of measurement data via communication networks or other means to a distant device where they are further processed and/or used for legally regulated purposes.

3.2.28 Type-specific parameter

Legally relevant parameter with a value that depends on the type of instrument only. Type-specific parameters are part of the legally relevant software.

3.2.29 User interface

An interface that enables information to be interchanged between a human user and the measuring instrument or its hardware or software components, as eg. switches, keyboard, mouse, display, monitor, printer, touch-screen, a software window on a screen including the software that generates it.

3.3 Validation and Verification Terminology

3.3.1 Acceptable solution

A design or a principle of a software module or hardware unit, or of a feature that is considered to comply with a particular requirement. An acceptable solution provides an example of how a particular requirement may be met. It does not prejudice any other solution that also meets the requirement.

3.3.2 Conformity of software

Degree of analogy of software in production line measuring instrument with the approved software (at type approval).

3.3.3 Sealing

To set a special protection to serve as indicator for case of unauthorised access to the device's hardware or software part.

3.3.4 Securing

To prevent unauthorised access to the device's hardware or software part.

3.3.5 Software test

Technical operation that consists of determination of one or more characteristics of a software according to the specific procedure (analysis of technical documentation or running the programme under controlled conditions).

3.3.6 Software validation [FDA General Principle of Software Validation, clause 3.1.2]

Confirmation by examination and provision of objective evidence that software specification conform to the user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled

3.3.7 Validation [similar ISO/IEC 14598, clause 4.24 and IEC 61508-4, clause 3.8.2]

Confirmation by examination and provision of objective evidence (i.e. information that can be proved true, based on facts obtained from observations, measurement, test, etc.) that the particular requirements for the specific intended use are fulfilled. In the present case the related requirements are those of this Recommendation.

3.3.8 Verification [VIML, 2.13]

Procedure (other than type approval) which includes the examination and marking and/or issuing of a verification certificate that ascertains and confirms that the measuring instrument complies with the statutory requirements. [6])

3.4 Abbreviations

EUT	Equipment Under Test
IEC	International Electrotechnical Committee
I/O	Input / Output (refers to ports)
ISO	International Organisation for Standardisation
IT	Information Technology
MPE	Maximum Permissible Error
N.A.	Not applicable
OIML	International Organisation of Legal Metrology

4 Instructions for use of this Document in drafting OIML Recommendations

4.1 Provisions of this document apply only to new OIML Recommendations and OIML Documents under revision.

4.2 All normative 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 normative documents.

4.3 It is the objective of this Document to provide the TCs or SCs responsible for elaboration of 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 TC or SC shall determine which level for protection or conformity issues or validation intensity is suitable and how to incorporate the relevant portions of this document into their Recommendation. In Chapter 8 some aid is given for performing this task.

---

[6]) Note: Differing definition from other standards like e.g. ISO/IEC 14598, clause 4.23 or IEC 61508-4, clause 3.8.1.

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

The TC's and SC's should use this guidance document to establish software related requirements in addition to the other technical and metrological requirements of the relevant Recommendation.

5.1 General requirements

At the time of publication of 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 instruments, electronic devices and sub-assemblies and should be considered in all OIML Recommendations. In contrast to these elementary requirements the specific ones (5.2) deal with technical feature that are not common for some kinds of instruments or in some areas of application.

Notation: (I) – *Technical solution acceptable in case of normal severity level*

(II) – *Technical solution acceptable in case of raised severity level (see 8)*

5.1.1 Software identification

Requirement: Legally relevant software [[Software under metrological control ???]] shall be clearly identified with the software version or another token. The identification may consist of more than one part but one part shall be only dedicated for the legal purpose.

The identification shall be inextricably linked to the software itself and shall be presented at start-up, on command or during operation on the display. If a sub-assembly has no display, the identification shall be sent via communication interface in order to be displayed on another sub-assembly.

Purpose: 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 conform.

Example: (I) The software contains a textual string of a number or other characters 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.

(II) The software calculates a checksum of the executable code and presents the result as the identification instead of or additional to the string in (I). The CRC16 algorithm is an acceptable solution for this calculation.

Solution (II) is suitable, if increased conformity is required (see 5.2.5, (d) and 8).

5.1.2 Correctness of algorithms and functions

Requirement: The measuring algorithms and functions of a measuring device shall be correct (accuracy of the algorithms like filtering of A/D conversion results, price calculation according to certain rules, rounding algorithms, ...).

The measurement result and accompanying information required by specific Recommendations or 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).

5.1.3 Software protection

5.1.3.1 Prevention of accidental misuse

Requirement: A measuring instrument – especially the software – shall be constructed in a way that possibilities for unintentional accidental or intentional misuse are minimal.

Purpose: Software controlled instruments are often complex in their functionality. The user needs good guidance for correct use and for achieving correct measurement results. The presentation of the measurement results should be unambiguous for all parties affected.

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

5.1.3.2 Fraud protection

Requirement (a): Metrologically critical software shall be secured against inadmissible modification, loading, or changes by swapping the memory device.

Note (a): This requirement implies that technical means – not only mechanical sealing – are necessary for computers as part of a measuring instrument having an operating system or an option to load software.

Example (a): (I)/(II) The housing containing the memory devices is sealed or the memory device is sealed on the PCB[7]).

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

(I) The metrologically crucial software is residing on a device or sub-assembly that can be mechanically sealed. Some metrological functions are relocated to a general purpose computer with an operating system. Swapping this software part is inhibited by simple cryptographic means, eg. encryption of the data transfer between the sub-assembly and the general purpose computer. The key for decryption is hidden in the legally relevant programme of the general purpose computer. Only this programme knows the key and is able to read, decrypt and use the measurement values. Other programmes cannot be used for this purpose as they cannot decrypt the measurement values (see also 5.2.1.2 (c)).

Requirement (b): If commands can be entered via a user interface, they shall be described completely in the software documentation to be submitted for the type approval. Only documented functions are allowed to be activated by the user interface. The user interface shall be realised in a way that it does not facilitate fraudulent use. The presentation of information shall comply with 5.2.2.

Note (b): The examiner decides whether all of these documented commands are acceptable.

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

Requirement (c): Parameters that fix legally relevant characteristics of the measuring instrument shall be secured against unauthorised modification. The current parameter settings must be able to be displayed or printed.

Note (c): Device-specific parameters may be adjustable or selectable only in a special operational mode of the instrument. They may be classified as those that should be secured (unalterable) and those that may be accessed (*settable* parameters) by an authorised person, e.g. instrument owner or product vendor.

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

---

[7]) PCB – Printed Circuit Board

Example (c): (I)/(II) Device-specific parameters to be secured are stored in a non-volatile memory. Its write-enable input is inhibited by a switch that can be sealed. The circuit is designed in a way that the write-protection cannot be cancelled by a short-circuit of contacts.

(I)/(II) Settable parameters are stored in another non-volatile memory. Its write-enable input may be software-controlled. The software allows programming of this memory if the user has input a correct password.

Requirement (d): Protection comprises mechanical sealing and electronic or cryptographic means making an inadmissible intervention impossible or evident. [(R 105 – 13.3.1a, R 117 – 4.3.3.1a, R 74 – 3.4.2) Ref. to be removed in final version]

Example (d): See (a).

Level (II) of the examples for acceptable technical solutions is appropriate, if increased protection against fraud is necessary (see 8).

5.1.4 Support of hardware features

5.1.4.1 Support of fault detection

The TC or SC responsible for particular Recommendation may require fault detection for certain failures (addressed in D11 (5.1.2 (b) and 5.3)). The manufacturer is free to design checking facilities in software or hardware or let hardware facilities be supported by software.

Requirement: If software is involved in fault detection, an appropriate reaction is required. The responsible TCs may prescribe that the instrument / device is deactivated or an alarm / report is generated in case a fault condition is detected.

The documentation submitted for type approval shall contain a list of faults that are detected by the software and if necessary for understanding, a description of the detecting algorithm.

Example: (I)/(II) On each start-up the legally relevant programme calculates a checksum of the programme 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 don't match, the programme stops execution.

If the measurement is not interruptible the checksum is calculated cyclically controlled by a software timer. In case a failure is detected, the software displays an error message or switches on a failure indicator and registers the time of the event in a log if it exists.

An acceptable checksum algorithm is CRC16.

5.1.4.2 Support of durability protection

It is the manufacturer's choice to realise durability protection facilities addressed in D11 (5.1.3 (b) and 5.4) in software or hardware or let hardware facilities be supported by software. The TC or SC responsible for particular Recommendation may recommend appropriate solutions.

Requirement: If software is involved in durability protection, an appropriate reaction is required. The responsible TCs may prescribe that the instrument / device is deactivated or an alarm / report is generated in case durability is detected being jeopardised.

The documentation shall contain a list of durability errors that are detected by the software and if necessary for understanding, a description of the detecting algorithm.

Example: (I)/(II) An exhaust gas analyser needs a re-calibration after a certain time interval for guaranteeing durability of measurement. The software gives a warning when the maintenance interval has elapsed and even stops measurement, if it has been exceeded for a certain amount.

5.2 Requirements specific for Configurations

The requirements given in this section are based on typical technical solutions in IT though 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 used in measuring systems. They have to be considered in addition to those described in 5.1.

Notation: (I) – *Technical solution acceptable in case of normal severity level*

(II) – *Technical solution acceptable in case of raised severity level (see 8)*

5.2.1 Specifying and separating relevant parts and specifying interfaces of parts

Requirement: Metrologically critical parts of a measuring system – whether software or hardware parts – shall not be inadmissibly influenced by other parts of the measuring system.

This requirement applies, if the measuring instrument (or device or sub-assembly) has interfaces for communicating with other devices or if there are other software parts besides the metrologically critical parts within a measuring instrument (or device or sub-assembly).

5.2.1.1 Separation of devices and sub-assemblies

Requirement (a): Sub-assemblies or electronic devices of a measuring system that perform legally relevant functions [[functions under metrological control ???]] shall be identified, clearly defined, and documented. They form the legally relevant part of the measuring system.

Note (a): The examiner decides whether this part is complete and whether other parts of the measuring system may be excluded from further evaluation.

Requirement (b): It it shall be shown that the relevant functions and data of sub-assemblies and electronic devices cannot be inadmissibly influenced by commands received via the interface.

This implies that there is an unambiguous assignment of each command to all initiated function or data change in the sub-assembly or device. The commands and their effects shall be described completely in the software documentation to be submitted for type approval. Signals or codes that are not declared and documented as commands shall have no effect on the sub-assembly's or device's functions and data. The manufacturer shall state the completeness of the documentation of commands.

Note (b): Commands may be a sequence of electrical (optical, electromagnetic, etc.) signals on input interfaces or codes in data transmission protocols that may be accompanied with additional data.

If "legally relevant" sub-assemblies or devices interact with other "legally relevant" sub-assemblies or devices, refer to 5.2.3.

Example (a)/(b): (I)/(II) An electricity meter is equipped with an optical interface for connecting a device to read out measurement values. The meter stores all 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 device.

The software of the electricity meter is able to receive commands for selecting the quantities wanted. It combines the measurement value with additional information – eg. timestamp, unit – and sends this data set back to the requesting device. The software only accepts commands for selection of valid allowed quantities and discards any other command sending back only an error message.

Inside the housing that can be sealed there is a switch that defines the operating mode of the electricity meter: one switch setting indicates the verified mode the other the non-verified mode to the software. In the non-verified mode the command set is extended compared to the mode described above; eg. it may be possible to adjust the calibration factor by a command that is discarded in the verified mode.

5.2.1.2 Separation of software parts

Requirement (a): All software modules (programmes, subroutines, objects etc.) that perform legally relevant functions or that contain legally relevant data domains form the legally relevant software part of a measuring instrument (device or sub-assembly). The conformity requirement applies to this part (see 5.2.5) 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 (a): (I) A measuring system consists of several load cells 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 realising legally relevant functions into a dynamically linkable library. One or several legally non-relevant applications may call programme procedures in this library. These procedures receive the measurement data from the load cells, calculate the measurement result, and display it in a software window. When having finished the legally relevant functions, control is given back to the legally non-relevant application.

Requirement (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. This implies that all legally relevant functions and data domains of the software are described to enable a type approval authority to decide on correct software separation.

The interface consists of programme 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 programme code is part of the software interface. The data domain forming the software interface including the code that exports from the legally relevant part to the interface data domain and the code that imports from the interface to the legally relevant part shall be clearly defined and documented. The declared software interface shall not be circumvented.

There shall be an unambiguous assignment of each command to all initiated function or data change in the legally relevant part of the software. Commands that are not declared and documented as commands shall have no effect on the legally relevant part of the software. The manufacturer shall state the completeness of the documentation of commands.

Note (b): Commands may be a sequence of data that causes the legally relevant software part to perform certain functions or data changes.

Example (b): (I) In the example described in (a) the software interface is realised by the parameters and return values of the procedures in the library. No pointers to data domains inside the library are returned. The definition of the interface is fixed in the compiled legally relevant library and cannot be changed by any application. It is not impossible to circumvent the parameters and address data domains of the library directly; but this is no good programming practice, is rather complicated, and may be classified as hacking.

Requirement (c): Software separation implies that if the system has limited resources, the legally relevant software has priority over the legally not relevant software. The measurement task (realised by the legally relevant software part) must not be delayed or blocked by other tasks.

Note: The manufacturer is responsible for respecting these constraints. Technical means (like sealing) for preventing a programmer from circumventing the interface or programming hidden commands are not possible. The programmer of the legally relevant software part should be instructed by the manufacturer about these requirements.

Example (c): (I) In the example (a)/(b) 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 taken in the example system to fulfil the requirement (c): The load cells 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. If the application programmer wants to read and process measurement values, he is forced to use the legally relevant procedures in the library that perform all legally required functions as a side effect when being called. The library contains procedures that export the decrypted measurement values allowing the application programmer to use them for his own needs *after the legally relevant processing has been finished*.

Examples (a) to (c) are acceptable as a technical solution only for a normal severity level (I). If increased protection against fraud or increased conformity is necessary (see 8), software separation as described is ***not*** *acceptable. In this case the software should be subject to legal control as a whole.*

5.2.2 Shared indications

A display or printout may be used for presenting both information from the legally relevant part of software and other information. In that case the following requirement applies:

Requirement: The distinction between these information shall be clear and unambiguous. (OIML R 125 – 7.1, 7.6e)[To be removed in the final version]. A specific indication may support this feature.

If remote indications are allowed in the area of application (to be defined by the responsible TCs), a clear assignment to the measuring instrument or place of measurement shall be provided.

Example: (I) On a printout of a fuelling system the lines containing the measurement values are marked by asterisks. The meaning is explained to the customer on each ticket.

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

(I) On a system described in 5.2.1.2, examples (a) to (c) the measurement values are displayed in a separate software window. The means described in (c) guarantee that only the legally relevant programme part can read the measurement values. On a windows based operating system an additional technical means is taken to meet the requirement in 5.2.2: 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 other windows that currently exist and bring it on top, if not.

The use of an off-the-shelf general purpose computer is not appropriate as part of a measuring system if increased protection against fraud is necessary (II). Additional hardware components are necessary to guarantee a sufficient level of protection.

5.2.3 Storage of data, transmission via communication systems

If measurement values are used at another place than the place of measurement or at a later time than the time of measurement they possibly have to leave the measuring instrument (device, sub-assembly) and be stored or transmitted in an insecure

environment before they are used for legal purposes. In this case the following requirements apply:

Requirement (a): The measurement value stored or transmitted shall be accompanied by all relevant information necessary for future legally relevant use. (OIML R 117 - 3.5.5).

Example (a): (I)/(II) A data set consists eg. of the following entries:

- measurement value including unit
- timestamp of measurement
- place of measurement or identification of the measuring instrument that was used for the measurement
- unambiguous identification of the measurement eg. consecutive numbers enabling assignment to values printed on an invoice.

Requirement (b): The data shall be protected by software means to guarantee authenticity, integrity and, if necessary correctness of the information of the time of measurement. The software that displays or further processes the measurement values and accompanying data shall check time of measurement, authenticity, and integrity of the data after having read them from the insecure storage or after having received them from an insecure transmission channel. If an irregularity is detected, the data shall be discarded or marked unusable (OIML R 117 – 4.3.5, R 49 - 4.3.3.1 and 4.3.3.2).

Note (b): Software modules that prepare data for storing or sending, or that check data after reading or receiving belong to the legally relevant software part.

Example (b): (I) The programme of the sending device calculates a checksum of the data set (algorithm CRC16) 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 stored as a constant in the programme code. The receiving or reading programme also has stored this initial value in its programme code. Before using the data set the receiving programme calculates the checksum and compares it with that stored in the data set. If both values match, the data set is not falsified. Else the programme assumes a falsification and discards the data set.

Requirement (c): For a high protection level it is necessary to apply cryptographic methods. Confidential keys used for that purpose shall be kept secret and secured in the measuring instruments, devices, or sub-assemblies involved. Means shall be provided that these keys can only be input or read if a seal is broken.

Example (c): (II) The storing or sending programme generates an “electronic signature” by first calculating a hash value[8]) and secondly encrypting the hash value with the secret key of a public key system[9]). The result is the signature. It is appended to the stored or transmitted data set. The receiver also calculates the hash value of the data set and decrypts the signature appended to the data set with the public key. The calculated and the decrypted values of the hash value are compared. If they are equal, the data set is not falsified (the integrity is proven). To prove the origin of the data set the receiver must know whether the public key really belongs to the sender ie. the sending device. Therefore the public key is displayed on the display of the measuring instrument and can be registered once eg. together with the serial number of the device when it is legally verified in the field. If the receiver is sure that he used the correct public key for decryption of the signature, also the authenticity of the data set is proven.

Choose level (II) of the example for acceptable technical solutions, if increased protection against fraud is necessary (see 8).

5.2.3.1 Automatic storing

Requirement: The measurement data must be stored automatically when the measurement is concluded, ie. when the final value used for the legal purpose has been generated. The long-term storage must have a capacity which is sufficient for the intended

---

[8]) Acceptable algorithms: SHA-1, MD5, RipeMD160

[9]) Acceptable algorithms: RSA (1024 bit key length), Elliptic Curves (160 bit key length)

purpose. (OIML R 117 - 3.5.2, R 49 - 4.3.3.1, 4.3.3.2). When the storage is full, it is permitted to delete memorised data when both of the following conditions are met:

- data shall be deleted in the same order as the recording order and the rules established for the particular application are respected,
- deletion shall be carried out after a special manual operation. (OIML R 117 - 3.5.3).

Note: Cumulative measurement values like eg. electrical energy or gas volume have to be updated currently. As always the same data domain (programme variable) is used the requirement concerning the storage capacity is not applicable to cumulative measurements.

5.2.3.2 Transmission delay

Requirement: The measurement shall not be inadmissibly influenced by a transmission delay. If network services become unavailable, no measurement data must get lost.

Example: (I)/(II) The sending device waits until the receiver has sent an affirmation of correct receipt of the data set. The sending device keeps the data set in a buffer until this affirmation has been received. The buffer may have a capacity for more than one data set, organised as a FIFO[10]) queue.

5.2.4 Compatibility of operating systems and hardware, portability

Requirement: The manufacturer shall identify the hardware and software environment that is suitable. Minimal resources and a suitable configuration (processor, RAM, HDD, specific communication, version of operating system...) which is necessary for correct functioning, has to be declared by the manufacturer. Technical means shall be provided in the legally relevant software to prevent operation, if the minimal configuration requirements are not met.

If correct functioning is only guaranteed in an invariant environment, means shall be provided to keep the environment fixed. This especially applies to universal computer performing legally relevant functions. It is in general necessary to fix hardware, operating system, or system configuration of a universal computer or even exclude the usage of an off-the-shelf universal computer in the following cases:

- if high conformity is required (see 5.2.5 (d)),
- if fixed software is required (eg. **5.2.6.2.2** for traced software update),
- if cryptographic algorithms or keys have to be implemented (see 5.2.3).

5.2.5 Conformity of production-line devices with the approved type

Requirement: The manufacturer shall produce devices and the legally relevant software that conform to the approved type and the documentation submitted. There are different levels of conformity demands:

(a) identity of the *legally relevant functions* described in the documentation (6.1) of each device with those of the type (the executable code may differ),

(b) identity of *parts of the legally relevant source code*, and the rest of the legally relevant software complying with (a),

(c) identity of the *whole legally relevant source code*, and

(d) identity of the *whole executable code*.

---

[10]) FIFO: First in – first out

It has to be defined for each kind of instrument or area of application by the responsible TCs which degree of conformity is suitable. The TCs could define a subset from these conformity degrees for a particular kind of instrument and leave the decision what degree of conformity is to be applied to the approving body.

Except for (d) there may be a software part with no conformity requirements, if it is separated from the legally relevant part according to 5.2.1.2.

Means described in 5.1.1 and 5.2.1 shall be provided to make the conformity evident.

5.2.6 Maintenance and re-configuration

Requirement: Only versions of legally relevant software that conform with the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also on the kind of instrument under consideration. The following options 5.2.6.1 and 5.2.6.2 are equivalent alternatives. This issue concerns verification in the field. Refer to chapter 7 for additional constraints.

5.2.6.1 Verified update

The software to be updated can be loaded locally ie. directly on the measuring device or remotely via a network. Loading and installation may be two different steps (as shown in Fig. 5-1) or combined to one, depending on the needs of the technical solution. After 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 used for legal purposes before a (subsequent) verification of the instrument as described in chapter 7 has been performed and the securing means have been renewed (if not otherwise stated in the relevant OIML Recommendation or in the approval certificate). A person responsible for verification must be at place.

5.2.6.2 Traced update

The software is implemented into the instrument according to the requirements for traced update (**5.2.6.2.1** to **5.2.6.2.6**) if it is in compliance with the relevant OIML Recommendation. Traced update is the procedure of changing software in a verified instrument or device after which the subsequent verification by a responsible person at place is not necessary. The software to be updated can be loaded locally ie. directly on the measuring device or remotely via a network. The software update is recorded in an audit trail (see **5.2.6.2.5**). The procedure of a traced update comprises several steps: loading, integrity checking, checking of the origin (authentication), installation, logging and activation.

5.2.6.2.1 Traced update of software shall be automatic. On completion of the update procedure the software protection environment shall be at the same level as required by the type approval.

5.2.6.2.2 The target measuring instrument (device, sub-assembly) shall have a fixed legally relevant software that cannot be updated and that contains all of the checking functions necessary for fulfilling traced update requirements.

5.2.6.2.3 Technical means shall be employed to guarantee the authenticity of the loaded software ie. that it originates from the owner of the type approval certificate. This can be accomplished eg. by cryptographic means like signing. The signature is checked during loading. If the loaded software fails this test, the instrument shall discard it and use the previous version of the software.

5.2.6.2.4 Technical means shall be employed to guarantee the integrity of the loaded software ie. 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. If the loaded software fails this test, the instrument shall discard it and use the previous version of the software

5.2.6.2.5 It shall be guaranteed by appropriate technical means eg. an audit trail that traced updates of legally relevant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection. 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 (that depends on national legislation).

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

The traceability means and records are part of the legally relevant software and should be protected as such. The software used for displaying the audit trail belongs to the fixed legally relevant software.

5.2.6.2.6 It shall be guaranteed by technical means that software may only be updated with the explicit consent of the user or owner of the measuring instrument. Relevance of this requirement depends on national legislation.

5.2.6.2.7 If the requirements **5.2.6.2.1** to **5.2.6.2.6** 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 5.2.1.2.
- The whole legally relevant software part cannot be updated without breaking a seal.
- It is stated in the type approval certificate that updating of the legally non-relevant part is acceptable.

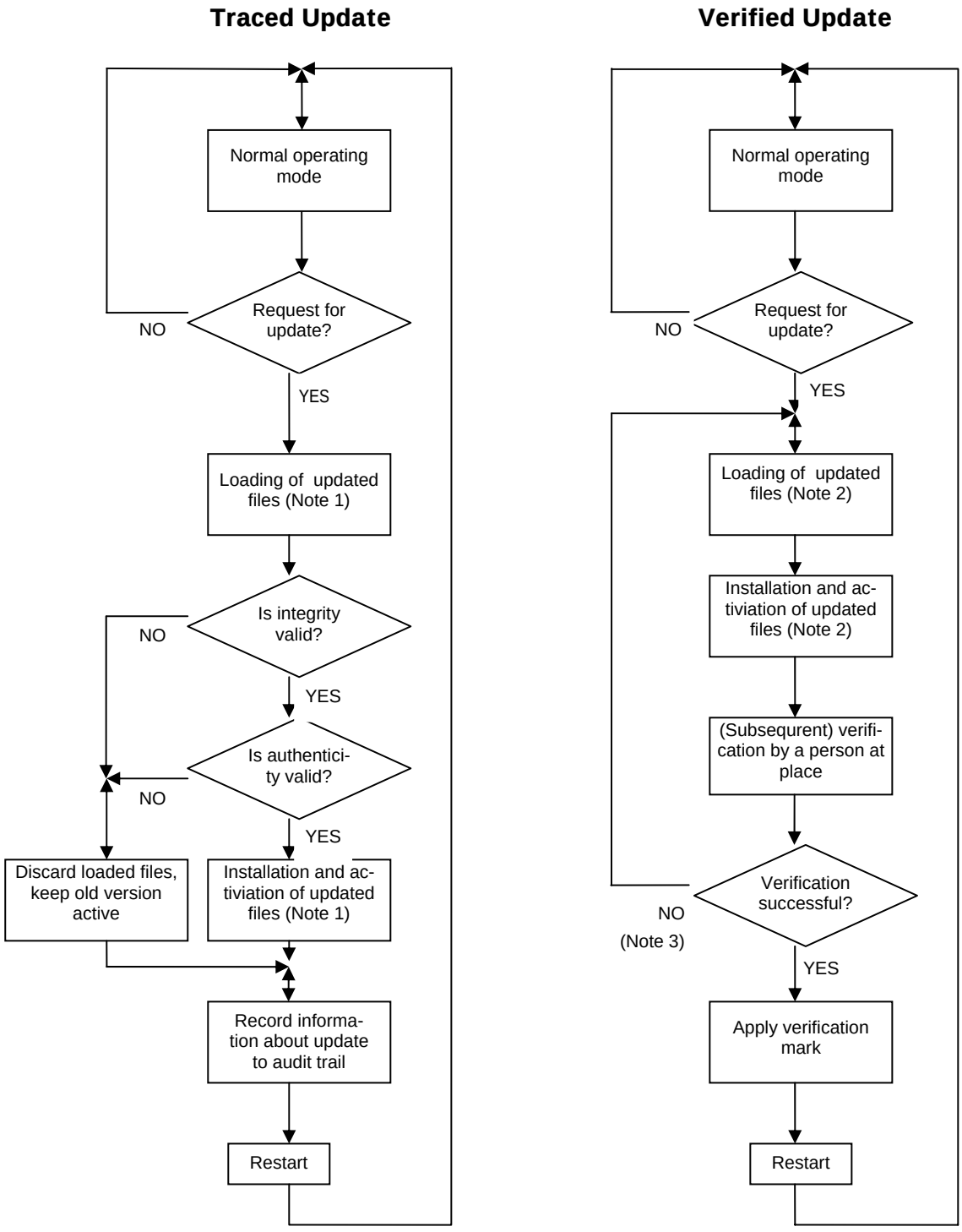

Figure 5-1: Software update procedures

Notes to Figure 5-1**:**

1) In case of *Traced update* updating is separated into the 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 fall back to the old version, if the checks fail.
2) In case of *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 failing of the verification because of the software update is considered. Failing because of other reasons doesn’t require re-loading and re-installing of the software, symbolised by the NO-branch.

6 Type approval

6.1 Documentation to be supplied for type approval

For type approval the manufacturer of the measuring instrument shall declare and document all programme functions, relevant data structures and interfaces that are implemented in the instrument. There shall not exist any hidden undocumented functions.

6.1.1 Typical documentation (for each measuring instrument, device, or sub-assembly) basically include:

- A description of the legally relevant software
  - List of software modules that belong to legally relevant part (annex C) including a declaration that all functions that have an influence on the measurement are included in the description
  - Description of the software interfaces (annex C).
  - Description of the generation of the software identification
  - Depending on the validation method chosen by the responsible TCs (see 6.4) the source code shall be made available to the testing authority
  - List of parameters to be protected and description of protection means
- A Description of minimal system configuration (see 5.2.4)
- A Description of security means of the operating system (password, ... if applicable)
- A description of the accuracy of the algorithms (like filtering of A/D conversion results, price calculation, rounding algorithms, ...).
- 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 interface including statement of completeness
- A description of data sets stored or transmitted.
- If fault detection is realised in software, a list of faults that are detected and a description of the detecting algorithm.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network etc.
- The operating manual.

Furthermore, the application for type approval 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 Recommendation, in which the general requirements of this Document have been incorporated.

6.2 Requirements on the approval procedure

Test procedures in the framework of the type approval eg. described in Document D11 are based on well defined test setups and test conditions and can rely on precise comparative measurements. “Testing” and “validating” software means something different. The accuracy or correctness of software in general cannot be measured in a metrological sense though there are standards how to “measure” software quality [eg. ISO/IEC 14598]. The procedures described here take into consideration both the needs in legal metrology and well-known validation and test methods in software engineering not having the same goal like eg. the software developer who is searching errors and optimising performance. As shown in 6.4 each software requirement needs

individual adaptation of suitable validation procedures. The effort for the procedure should reflect the importance of the requirement in terms of accuracy, reliability and protection against corruption.

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

Methods described in the following focus on the type examination. Verifications of every single instrument in use in the field are not covered.

The methods proposed for software validation are described in 6.3. Combinations of these methods forming a complete validation programme adapted to all requirements defined in Section 0 are specified in Chapter 6.4.

6.3 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.

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 features (6.3.2.2)	Correctness of the algorithms, uncertainty, compensating and correcting algorithms, rules for price calculation	Documentation	-
VFTSw	Validation by functional testing of software features (6.3.2.3)	Handling by the user, correct functioning of communication, indication, fraud protection, protection against operating errors	Documentation, text editor	-
DFA	Metrological data flow analysis (6.3.2.4)	Software separation, evaluation of the impact of commands on the instrument's functions	Source code, text editor (simple procedure), tools (sophisticated procedure)	Knowledge of programming languages. Instruction for the method necessary.
CIWT	Code inspection, Walkthrough (6.3.2.5)	All purposes	Source code, text editor	Knowledge of programming languages, protocols, and other IT issues
SMT	Software module testing (6.3.2.6)	All purposes when input and output can clearly be defined	Source code, testing environment, special software tools	Knowledge of programming languages, protocols, and other IT issues. Instruction for using the tools necessary.

Table 6-1: 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 This is the basic procedure that has to be applied in any case.

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

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

b) 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.

Concerning interfaces the documentation shall provide a complete list of commands or signals that the software is able to interpret. The effect of each command shall be documented in detail. It shall be described how the instrument reacts on undocumented commands.

c) Additional documentation of the software for complex measuring algorithms, cryptographical functions, or crucial timing constraints should be provided, if necessary for understanding and evaluating the software functions.

d) When it is not clear how to validate a function of a software programme 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.

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

Description The examiner tries to understand the functions and features of the measuring instrument using the verbal description and graphical representations and decides whether they comply with the requirements of the relevant Recommendation. Metrological requirements as well as software-functional requirements defined in chapter 0 (like eg. fraud protection, protection of adjustment parameters, disallowed functions, communication with other devices, update of software, fault detection) have to be considered and evaluated. This task may be supported by checklists (see Annex D).

Result The procedure gives a result for all characteristics of the measuring instrument provided an appropriate documentation has been submitted by the manufacturer. The result should be documented in a test report (see Annex C) included in the Test Report Format of the relevant Recommendation.

Complementing procedures Additional procedures should be applied, if examining the documentation cannot give substantiated validation results. In most cases "Validating the metrological functions by functional testing" (see 6.3.2.2) is a complementing procedure.

References FDA, General Principles of Software Validation; Final Guidance for Industry and FDA Staff, 11 January 2002

FDA, Guidance for FDA Reviewers and Industry, 29 May 1998

IEC 61508-7, 2000-3

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

Application Correctness of algorithms for calculating the measurement value from raw data, for linearisation of a characteristic, compensation of environmental influences, rounding in price calculation etc.

Preconditions Operating manual, functioning pattern, metrological references and test equipment.

Description Most of the approval and test methods described in Recommendations are based on reference measurements under varying conditions. Its application is not restricted to a certain technology of the instrument. Though it doesn't aim primarily on validation of 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 Recommendation cover all 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 Correctness of algorithms OK or not. Measurement values under all conditions within MPE or not.

Complementing procedures The method is normally an enhancement to 6.3.2.1. In certain cases it may not be possible 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) eg. for dynamic measurements.

References Various specific OIML Recommendations.

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

Application Validation of eg. 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 pattern, test equipment.

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 eg.

- Normal operating of the instrument if operating is software controlled. All switches or keys and described combinations should be used and the reaction of the instrument be 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 check whether this is detected by the programme.
- Generation and indication of a 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 check the correct reaction of the instrument.
- If configuration or environment of the legally relevant software is claimed to be fixed, protection means can be checked by making inadmissible changes. The software should inhibit these changes or should stop.

Result Software controlled feature under consideration OK or not.

Complementing 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 inadmissible commands only by trying commands at random. Apart from that a sender is needed to generate these commands. For normal validation level method 6.3.2.1 including a declaration of the manufacturer may cover this requirement. For extended examination level a software analysis like 6.3.2.4 or 6.3.2.5 is necessary.

References WELMEC 2.3, 7.2, FDA Guidance for Industry Part 11, August 2003

6.3.2.4 Metrological Dataflow Analysis (DFA)

Application Construction of the flow of measurement values through the data domains subject to legal control. Examination of the software separation.

Preconditions Software documentation, source code, editor, text search programme 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 is searched that reads them. This subroutine will store them in a variable after possibly having done some calculation. 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 transporting these values can be found in the source code simply by using a text editor and a text search programme for finding variable or subroutine names in another source code file than the currently opened in the text editor.

Other data flows can be found by this method eg. 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 5.2.1.2 is OK or not.

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

References 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, tries to understand the respective part of the code and decides whether the requirements are fulfilled and whether programme 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 analysing and checking.

Prior to these examination steps he usually has identified the legally relevant software part eg. by applying the metrological data flow analysis (see 6.3.2.4). In general code inspection or walk through is limited to this part. By combining both methods the examination effort is minimal compared to the application of these methods in the normal soft-

ware production with the objective of producing failure-free programmes or optimising performance.

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

Complementing 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.

References IEC 61508-7

6.3.2.6 Software Module Testing (SMT)

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

Preconditions Source code, development tools (at least a compiler), functioning environment of the software module under test, input data set and corresponding correct reference output data set or tools for automation. Skill 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 ie. a specific test programme module that is calling the module under test and providing it with all necessary input data. The test programme receives output data from the module under test and compares them with the expected reference values.

Result Measuring algorithm or other tested functions correct or not.

Complementing procedures This is an enhanced method, additional to 6.3.2.1 or 6.3.2.5. It is only profitable in exceptional cases.

References IEC 61508-7

6.4 Validation programme

Validation procedure consists of a combination of analysis methods and tests. The relevant Recommendation may specify details concerning the validation programme, 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 C).

In Table 6-2 two alternative levels A and B for the validation procedures are defined. Level B implies an extended examination compared to A. A selection between A and B type of validation procedures may be made by the responsible TCs - different or equal for each requirement - in accordance to 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 accidental 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 of relevant parts and specifying of interfaces of parts			
5.2.1.1	Separation of devices and sub-assemblies	AD	AD	
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, transmission via communication systems	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select »B« if transmission of measurement data in open system is foreseen
5.2.3.1	The measurement data must be stored automatically when the measurement is concluded	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select »B« in case of high risk of fraud
5.2.3.2	Transmission delays	AD + VFTSw	AD + VFTSw + SMT	Select »B« in case of high risk of fraud, eg. transmission in open systems
5.2.4	Compatibility of operating systems and hardware, portability	AD + VFTSw	AD + VFTSw + SMT	
	Maintenance and reconfiguration			
5.2.6.1	Verified update	AD	AD	
5.2.6.2	Traced update	AD + VFTSw	AD + VFTSw + CIWT/SMT	Select »B« in case of high risk of fraud

Table 6-2: Recommendations for combinations of analysis and test methods for the various software requirements (acronyms defined in **Table 6-1**)

6.5 Equipment under test (EUT)

Normally tests will be 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 device (module) of the measuring instrument is concerned, the relevant Recommendation may indicate that the tests, or certain tests, shall be carried out on the electronic devices or software modules separately, provided that, in case of tests with the devices in operation, these devices are included in a simulated set-up, sufficiently representative of its normal operation.

7 Verification

If a 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 adjustment, the conformity to a type.

The procedure of software update is described in 5.2.6 and **Figure 5-1**.

((To be completed))

8 Assessment of severity (risk) levels

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

Moreover, this guide does not interfere with the liberty of the technical committees and subcommittees to provide for severity levels that differ from those resulting from the guidelines set forth in this Document. Different severity levels may be used in accordance with special limits prescribed in the relevant Recommendations.

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

(a) risk of fraud
- the consequence and the social and societal impact of errors
- the value of 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) the possibility to repeat a measurement or to interrupt it.

Throughout the requirements' section (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 also ex-

amples with enhanced counter measures are presented that consider a raised severity level of the aspects described above *(marked with (II))*.

{{ Possibly to be enhanced }}

9 Assessment of software processes

The correctness of the software, the protection against manipulations, etc. are issues that concern not only tests and examinations of type approvals. Rather they are a matter of the quality of software processes which take place during the design, development, maintenance, and use of software. There are several international standards dealing with the assessment of software processes.

Since this issue is judged by the TC5/SC2 members to be of lower importance, the elaboration is deferred for the time being.

{{To be elaborated when this Document is finished. }}

ANNEX A

NOTES and BIBLIOGRAPHY

(to be elaborated, similar to D11)

ANNEX B

Examples of acceptable technical solutions

(to be elaborated)

ANNEX C

SAMPLE TEST REPORT FORMAT

Note: The Technical Committees and Sub-Committees developing Recommendations should decide which information shall be included in Test Report and OIML Certificate of Conformity. Eg. 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 Dynaflow model DF100

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-SW, 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
Dynaflow
P.O. Box 1120333
100 Reykjavik
Iceland
Reference: Mr Bjarnur Sigfridson

Applicant
New Company
New Street 123
1000 Ljubljana
Slovenia
Reference: Janez Novak

Test Object

The Dynaflow flow meter DF100 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 long-term storage of legally relevant data.

The flow meter DF100 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

Dynaflow, P.O. Box 1120333, 100 Reykjavik, Iceland.

The executable file name is "**df100_12.exe**".

The validated version of this software is **V1.2c**.

The source code comprises 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 "**df100_12.exe**" is protected against modification by a checksum. The value of checksum by algorithm XYZ is **1A2B3C**.

The software version is presented on the display upon device start-up and by pressing the "level" button for 4 seconds.

The validation has been supported by following documents from the manufacturer:

- DF 100 User Manual Release 1.6
- DF 100 Maintenance Manual Release 1.1
- Software description DF100 (internal design document, dated 22 Nov 2003)
- Electronic circuit diagram DF100 (drawing no 222-31, date 15 Oct 2003)

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

Performance of validation

The validation has been performed according to the OIML D-SW (version 1.0). The validation was performed between 1 November and 23 December 2003. A design review was held on 3 December by Dr K. Fehler at Dynaflow head office in Reykjavik. Other validation work has been carried out at the National Testing & Measurement Lab by Dr K. Fehler and M. S. Problème.

Following requirements have been validated:

- Software identification,
- Correctness of algorithms and functions,
- Software protection,
- Prevention of accidental misuse,
- Fraud protection,
- Support of hardware features,
- Storage of data, transmission via communication systems.

Following validation methods have been 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

Following requirements of the OIML D-SW have been validated without finding faults:

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.

Two commands which were not initially described in the operator's manual were found. The two commands have been included in the operator's manual dated 10 December 2003.

A software fault which limited the month of February to 28 days also in leap year was found in software package V1.2b. This has been corrected in V1.2c.

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

Conclusion

The software of the Dynaflow DF100 V1.2c fulfils the requirements of the OIML R-xyz.

National Testing & Measurement Lab
Software Department

Dr. K.E.I.N. Fehler	M. S.A.N.S. Problème
Technical manager	Technical Officer

ANNEX D

SAMPLE CHECKLIST

§(D-SW)	Requirement	+	-	Remarks
5.1	**General Requirements**			
5.1.1	**Software identification** *Legally relevant software shall be clearly identified.*			
5.1.2	**Correctness of algorithms and functiuns** The measuring algorithms and functions of a measuring device shall be correct			
5.1.3	**Software protection**			
5.1.3.1	**Prevention of accidental misuse** A measuring instrument – especially the software – shall be constructed in a way that possibilities for unintentional accidental misuse are minimal			
5.1.3.2	**Fraud protection** Metrologically critically software shall be secured against inadmissible modification, loading, or changes by swapping of hardware memory			
5.1.4	**Support of hardware features**			
5.1.4.1	**Support of fault detection** It is the manufacturer's choice to realise checking facilities addressed in D11 (5.1.2 (b) and 5.3) in software or hardware or let hardware facilities be supported by software.			
5.1.4.2	**Support of durability protection** *It is the manufacturer's choice to realise durability protection facilities addressed in D11 (5.1.3 (b) and 5.4) in software or hardware or let hardware facilities be supported by software*			
5.2	**Specific requirements**			
5.2.1	**Specifying and searating relevant parts anspecifying interface of parts** *Metrologically critical parts of a measuring system – whether software or hardware parts – shall not be inadmissibly influenced by other parts of the measuring system*			
5.2.1.1	**Separation of devices and sub-assemblies** Interfaces of these "legally relevant" sub-assemblies and devices shall be clearly defined and documented to show that their relevant functions and data cannot be inadmissibly influenced by commands received via the interface			

5.2.1.2	**Separation of software parts** *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*			
5.2.2	**Shared indications** *The distinction between information from the legally relevant part of software and other information shall be clear and unambiguous*			
5.2.3	**Storage of data, transmission via communication system** *The data shall be protected by software means to guarantee their identity, correctness of the information of the time of measurement, authenticity, and integrity*			
5.2.3.1	*The measurement data must be stored automatically when the measurement is concluded. The long-term storage must have a capacity which is sufficient for the intended purpose*			
5.2.3.2	*The measurement must not be inadmissibly influenced by a transmission delay. If network services become unavailable, no measurement data must get lost*			
5.2.4	**Compatibility of operating system and hardware, portability** *The manufacturer of the metrologically relevant software shall identify the hardware and software environment that is suitable*			
5.2.6	**Maintenance and reconfiguration** *Only approved versions of legally relevant software are allowed for use*			
5.2.6.1	**Verified update** *After update of the legally relevant software of a measuring instrument it is necessary to perform a verification of the instrument and renew the securing means*			
5.2.6.2	**Traced update** *The software is implemented into the instrument according to the requirements for traced update (***5.2.6.2.1** *to* **5.2.6.2.7***) if it is in compliance with national legislation. Traced update is the procedure of changing software in a verified instrument or device after which the verification is not necessary*			

ANNEX E

REFERENCE BETWEEN ANSWERS TO THE QUESTIONNARIE (Febr. 2002)

AND THIS PRE-DRAFT

Question, Problem	Levels of Importance Number of Vote			Result	Chapter where the issue is addressed
	High	Middle	Low		
Correctness	10	6	3	Important	5.1.2
Accidental Misuse	11	7	2	Important	5.1.3.1
Fraud protection	17	2	-	Important	5.1.3.2
Storage & transmission of data	13	6	-	Important	5.2.3
Support of hardware reliability	6	7	6	Less important	5.1.4
Compatibility, portability	8	9	1	Important	5.2.4
Identification of parts, interfaces	11	4	4	Important	5.2.1
Documentation	9	7	3	Important	6
Conformity with approved type	10	8	1	Important	5.1.1 5.2.4
Maintenance and Re-configuration	9	8	2	Important	5.2.6
Verification, certification	5	11	3	Less important	7
Assessment of Software processes	2	10	7	Less important	N. A.

ANNEX F

REFERENCE BETWEEN

DRAFT MEASUREMENT CANADA SPECIFICATION (METROLOGICAL SOFTWARE), 8 August 2002

AND THIS PRE-DRAFT

Canadian Specification		Comment	Chapter in D-SW
	Does not apply to Built-for-purpose devices	D-SW applies to all software controlled instruments	-
1	Definitions	No complete coverage, but no contradiction	3
2	Design, Composition, Construction		5.2.4
3(1)	Only functions related to measurement process: Accurate measurement at start up		5.1.2, 5.2.4
3(2)	Also functions other than related to measurement process: Accurate measurement at the time of measurement transaction		5.2.1, 5.2.1.2
4	Alteration of setup parameters or code		5.1.3.1, 5.1.3.2 (a), 5.1.3.2 (c)
5	Protection against changes		5.1.3
6(1)	Completeness and integrity of transmitted data		5.2.3
6(2), 6(3)	Reaction on corrupted transmitted data		5.2.3, , 5.1.4.1
7	Minimum hardware and software environment		5.2.4
8, 9, 10	Notification of changes		5.2.5
11	Power failure detection	In D-SW general fault detection addressed, not only power failure	5.1.4.1, 5.1.4.2
12	Exemption of hardware that is not temperature sensitive	Not a software issue	-
13 a)	Display or print parameter settings		5.1.3.2 (c)
13 c), d)	Display or print model and approval number of the software		5.1.1

Canadian Specification		Comment	Chapter in D-SW
13 b), e) – h)	Display or print approval information and information specific for volumetric liquid meters or weighing instruments	Canadian Specifications more detailed than D-SW	-
14	Event logger, audit trail	The Canadian Specifications and US HB44 describe audit trails when there is unlimited access to the instrument. Thea are more detailed than D-SW.	5.2.6.2.5
15	Not normal measuring (trade) mode		5.2.2
16	Other system components		5.2.1.1
17	Compatibility of hardware		5.2.4
18	Indication of measurement information		5.1.2, 5.2.2
19, 20	Record of measurement information	Canadian Specifications centred to weighing instruments, D-SW more general	5.1.2, 5.2.3
21	Influence of other software		5.2.1.2
22	Environmental temperature limits	Not a software issue	-
23	Avoid loss of unmeasured commodity or service installation	No direct analogy	(5.1.4.1, 5.1.4.2)
24	Visual means of display		5.1.2, 5.2.2
25	Connection to communication networks		5.2.3

ANNEX G

REFERENCE BETWEEN

WELMEC 7.2 SOFTWARE GUIDE[11]

AND THIS PRE-DRAFT

MID-Software Requirements		Comment	Chapter in D-SW
P1, U1	Documentation		6.1
P2, U2	Software identification		5.1.1
P3, U3	Influence via user interfaces		5.1.3.2 (b)
P4, U4	Influence via communication interface		5.2.1.1
P5, U5	Protection against accidental or unintentional changes		5.1.3.1, 5.1.4
P6, U6	Programme protection against intentional changes		5.1.3.2 (a), 5.1.3.2 (d)
P7, U7	Parameter protection		5.1.3.2 (c)
U8	Software authenticity and presentation of results		5.1.3.2 (a), 5.2.2, 5.2.5, **5.2.6.2.7**
U9	Influence of other software		5.2.1.2, 5.2.4
L1, T1	Completeness of stored or transmitted data		5.2.3
L2, T2	Protection against accidental or unintentional changes		5.2.3
L3, T3	Integrity of data		5.2.3
L4, T4	Authenticity of stored or transmitted data		5.2.3
L5, T5	Confidentiality of keys		5.2.3
L6, T6	Retrieval of stored data, Handling of corrupted data		5.2.3
L7	Automatic storing		5.2.3.1
L8	Storage capacity and continuity		5.2.3.1
T7	Transmission delay		5.2.3.2
T8	Availability of transmission services		5.2.3.2
S1	Realisation of software separation		5.2.1.2

[11]) WELMEC 7.2 Software Guide, Issue 1, May 2005.
MID – European Measurement Instrument Directive 2004/22/EG.

MID-Software Requirements		Comment	Chapter in D-SW
S2	Mixed indication		5.2.2
S3	Protective software interface		5.2.1.2
D1	Download mechanism		**5.2.6.2.1**, **5.2.6.2.2**
D2	Authentication of downloaded software		**5.2.6.2.3**
D3	Integrity of downloaded software		**5.2.6.2.4**
D4	Traceability of legally relevant software download		**5.2.6.2.5**
D5	Download consent		**5.2.6.2.6**