WELMEC 7.2
2015

WELMEC

European Cooperation in Legal Metrology

Software Guide

(Measuring Instruments Directive 2014/32/EU[1])

WELMEC

European cooperation in legal metrology

WELMEC is a cooperation between the legal metrology authorities of the Member States of the European Union and EFTA.

This document is one of a number of Guides published by WELMEC to provide guidance to manufacturers of measuring instruments and to Notified Bodies responsible for conformity assessment of their products.

The Guides are purely advisory and do not themselves impose any restrictions or additional technical requirements beyond those contained in relevant EU Directives.

Alternative approaches may be acceptable, but the guidance provided in this document represents the considered view of WELMEC as to the best practice to be followed.

Published by:
WELMEC Secretariat

e-mail : secretary@welmec.org
Website : www.welmec.org

Contents

Foreword

The Guide in hand is based on the “Software Requirements and Validation Guide”, Version 1.00, 29 October 2004, developed and delivered by the European Growth Network “*MID-Software”.* The Network was supported from January 2002 to December 2004 by the EU commission under the contract number G7RT-CT-2001-05064.

The Guide is purely advisory and does not itself impose any restrictions or additional technical requirements beyond those contained in the MID. Alternative approaches may be acceptable, but the guidance provided in this document represents the considered view of WELMEC as to a good practice to be followed.

Although the Guide is oriented on instruments included in the regulations of the MID, the results are of a general nature and may be applied beyond.

The version 2015 considers the latest experience gained from the applications of the Guide.

[1] **Please note**: The 2015 version of the guide remains also valid for Directive 2004/22/EC.

Introduction

This document provides technical guidance for the application of the Measuring Instruments Directive (MID), especially for software-equipped measuring instruments. It addresses all those who are interested in the technical understanding of software-related requirements of the MID, in particular of the essential requirements in annex 1 of the MID. The level of detail is oriented on the needs of manufacturers of measuring instruments and of notified bodies (NB) which perform conformity assessments of measuring instruments according to module B.

By following the Guide, a compliance with the software-related requirements of the MID can be assumed. It can be further assumed that all notified bodies accept this Guide as a compliant interpretation of the MID with respect to software. To show how the requirements set up in this Guide are related to the respective requirements in the MID, a cross reference has been included in this guide as an annex (Chapter 12).

Latest information relating to the Guides and the work of WELMEC Working Group 7 is available on the web site www.welmec.org.

1 Terminology

The terminology explained in this section describes the vocabulary as used in this guide. References to a standard or to any other source are given, if the definition is completely or in essential parts taken from it.

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.

Authentication: Verification of the declared or alleged identity of a user, process, or device.

Authenticity: Property of being genuine and able to be verified and be trusted [4].

Basic configuration: Design of the *measuring instrument* with respect to the basic architecture. There are two different basic configurations*: built-for-purpose measuring instruments* and *measuring instruments using a universal computer.* The terms are accordingly applicable to *sub-assemblies*.

Built-for-purpose measuring instrument (type P): A *measuring instrument* designed and built specially for the task in hand. Accordingly the entire application software is constructed for the measuring purpose. For a more detailed definition refer to Chapter 4.1.

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

Communication interface: An electronic, optical, radio or other technical interface that enables information to be automatically passed between parts of *measuring instruments, sub-assemblies*, or external devices.

Device-specific parameter: *Legally relevant parameter* with a value that depends on the individual instrument. Device-specific parameters comprise calibration parameters (e.g. span adjustment or other adjustments or corrections) and configuration parameters (e.g. maximum value, minimum value, units of measurement, etc). They are adjustable or selectable only in a special operational mode of the instrument. Device-specific parameters may be classified as those that should be secured (unalterable) and those that may be accessed (settable parameters) when the instrument is in use.

Integrated storage: non-removable storage that is part of the measuring instrument, e.g. RAM, EEPROM, hard disk.

Integrity of data and software: Assurance that the data and software have not been subjected to any unauthorised changes while in use, transfer or storage.

IT configuration: Design of the *measuring instrument* with respect to IT functions and features. There are four IT configurations considered in this guide: *long-term storage of measurement data*, *transmission of measurement data*, *software download* and *software separation* (see also *Basic configuration*). The terms are accordingly applicable to *sub-assemblies*.

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.*

Legally relevant software: Part of software including type-specific *parameters* that fulfils functions, which are subject to legal control. All other software is called legally non-relevant. Measurement data generated by the instrument or processed by legally relevant software is separately treated and not considered a part of legally relevant software.

Legally relevant software identifier: Identifiers of the legally relevant software are called the *legally relevant software identifiers*

Long-term storage of measurement data: Storage used for keeping measurement data ready after completion of the measurement for later legally relevant purposes (e.g. the conclusion of a commercial transaction).

Measurement data: Measurement values generated or processed by measuring instruments and accompanied by physical units and other information, e.g. time stamps, that is connected to them on a regular basis that characterise them metrological.

Measuring instrument: Any device or system with a measurement function. The adjective "measuring" is omitted if confusions can be excluded. [MID, Article 4]

Measuring instruments using a universal computer (type U): *Measuring instrument* that comprises a general-purpose computer, usually a PC-based system, for performing legally relevant functions. A type U system is assumed if the conditions of a *built-for-purpose measuring instrument (type P)* are not fulfilled.

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*).

Operating System: A collection of software, and firmware elements that control the execution of computer programs and provide such services as computer resource allocation, job control, input/output control, and file management in a computer system [5].

Protective Software Interface: Interface between the legally relevant and legally non-relevant software, for protection conditions see section S3.

Risk class: Class of *measurement instrument* types with almost same risk assessments.

Software download: The process of automatically transferring software to a target *measuring instrument* or hardware-unit using any technical means from a local or distant source (e.g. exchangeable storage media, portable computer, remote computer) via arbitrary connections (e.g. direct links, networks).

Software identifier: A sequence of characters, that identifies the software. The identifier is logically considered a part of the software.

Software separation: The unambiguous separation of software into *legally relevant software* and legally non-relevant software. If no software separation exists, the whole software is to consider as legally relevant.

Sub-assembly: A hardware device (hardware unit) that functions independently and makes up a *measuring instrument* together with other sub-assemblies (or a measuring instrument) with which it is compatible [MID, Article 4].

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.

TEC: Type examination certificate.

Type-specific parameter: *Legally relevant parameter* with a value that is equal for all instruments of the type. A type-specific parameter is considered a part of the legally relevant software.

User interface: An interface forming the part of the instrument or measuring system that enables information to be passed between a human user and the measuring instrument or its hardware or software parts, as, e.g. switch, keyboard, mouse, display, monitor, printer, touch-screen.

Validation: 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 intended use are fulfilled. In the present case the related requirements are those of the MID.

The following definitions are rather specific. They are only used in some extensions and for risk classes D or higher.

Hash algorithm: Algorithm that compresses the contents of a data block to a number of defined length (hash code), so that the change of any bit of the data block leads in practice to another hash code. Hash algorithms are selected such that there is theoretically a very low probability of two different data blocks having the same hash code.

Signature algorithm: A cryptographic algorithm that encrypts (encodes) a hash code using an encoding *key* and that allows decoding of the encrypted hash code if the corresponding *decoding key* is available.

Key: An appropriate number or sequence of characters used to encode and / or decode information.

Public Key System (PKS): A pair of two different *keys*, one called the secret key and the other the public key. To verify *integrity* and *authenticit*y of information, the hash value of the information generated by a *hash algorithm* is encrypted with the secret key of the sender to create the signature, which is decrypted later by the receiver using the sender's public key.

Public Key Infrastructure (PKI): Organisation to guarantee the trustworthiness of a *public key system*. This includes granting and distributing digital certificates to all members that take part in the information exchange.

Certification of keys: The process of binding a public key value to an individual, organisation or other entity.

Electronic signature: A short code (the signature) that is unambiguously assigned to a text, data block or binary software file to prove the *integrity* and *authenticity* of data stored or transmitted. The signature is created using a *signature algorithm* and a secret *key.* Usually the generation of an electronic signature is composed of two steps: (1) first a *hash algorithm* compresses the contents of the information to be signed to a short value, and (2) then a signature algorithm combines this number with the secret key to generate the signature.

Trust Centre: An association that trustworthily generates, keeps, and issues information about the authenticity of public keys of persons or other entities, e.g. measuring instruments.

2 How to use this guide

This section describes the organisation of the guide and explains how to use it.

2.1 Overall structure of the guide

The guide is organised as a structured set of requirement blocks. The overall structure of the guide follows the classification of measuring instruments into basic configurations and the classification of so-called IT configurations. The set of requirements is complemented by instrument-specific requirements.

Consequently, there are three types of requirement sets:

1. requirements for two basic configurations of measuring instruments (called type P and U),
2. requirements for four IT configurations (called extensions L, T, S and D)
3. instrument-specific requirements (called extensions I.1, I.2, …).

The first type of requirements is applicable to all instruments. The second type of requirements concerns the following IT functions: long-term storage of measurement data (L), transmission of measurement data (T), software download (D) and software separation (S). Each set of these requirements is only applicable if the corresponding function exists. The last type is a collection of further, instrument-specific requirements. The numbering follows the numbering of instrument-specific annexes in the MID. The set of requirement blocks that may be applied to a given measuring instrument is schematically shown in Figure 2-1.

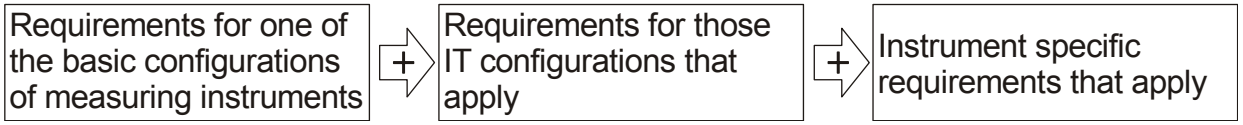

Figure 2-1: Type of requirement sets that should be applied to an instrument

The schemes in the following Figure 2-2 show what sets of requirements exist.

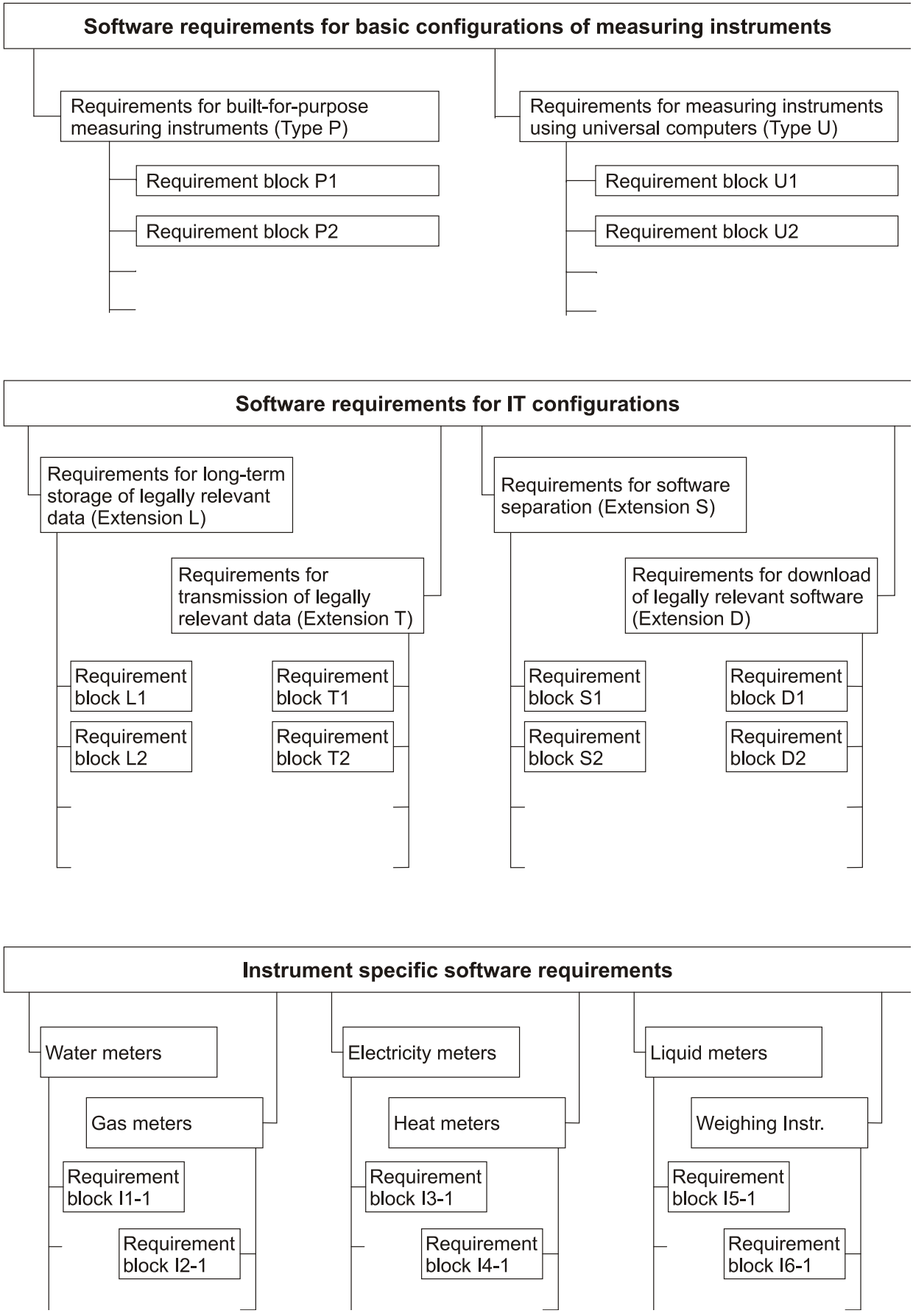

Figure 2-2: Overview of requirement sets

In addition to the structure described, the requirements of this guide are differentiated according to risk classes. Six risk classes, numbered from A to F with increasing risk assumptions, are introduced. The lowest risk class A and the highest risk classes E and F are not used for instruments under MID regulation, for the present. They are placeholders for the eventual case, that they will become necessary in future. The remaining risk classes B to D cover all of the instrument classes falling under the regulation of MID. Moreover, the risk classes from A to F provide a sufficient window of opportunity for the case of changing risk evaluations. The classes are defined in Chapter 3 of this guide.

Each measuring instrument shall be assigned to a risk class because the particular software requirements to be applied are governed by the risk class the instrument belongs to.

2.2 How to select the appropriate parts of the guide

This comprehensive software guide is applicable to a large variety of instruments. The guide is modular in form. The appropriate requirement sets can be easily selected by observing the following procedure.

Step 1: *Selection of the basic configuration (P or U)*

Only one of the two sets of requirements for basic configurations needs to be applied. Decide which basic configuration the instrument conforms to: a built-for-purpose instrument with embedded software (type P, see Chapter 4.1) or an instrument using a universal computer (type U, see Chapter 5.1). If not the whole instrument but only a sub-assembly of the instrument is the matter of concern, then decide accordingly for the sub-assembly. Apply the complete set of requirements that belongs to the respective basic configuration.

Step 2: *Selection of applicable IT configurations (extensions L, T, S and D)*

The IT configurations comprise: long term storage of measurement data (L), transmission of measurement data (T), software separation (S) and download of legally relevant software (D). The corresponding requirement sets, called modular extensions, are independent of each other. The sets selected depend only on the IT configuration. If an extension set is selected, then it shall be applied in full. Decide which, if any, of the modular extensions are applicable and apply them accordingly (Figure 2-2).

Step 3: *Selection of instrument specific requirements (extension I)*

Select - using the respective instrument specific extension I.x - which, if any, instrument specific requirements are applicable, and apply them accordingly (Figure 2-2).

Step 4: *Selection of the applicable risk class (extension I)*

Select the risk class as defined in the respective instrument specific extension I.x, sub-chapter I.x.6. There, the risk class is defined uniformly for a class of measuring instruments or possibly further differentiated for categories, fields of application, etc. Once the applicable risk class has been identified, only the respective requirements and validation guidance need to be considered.

2.3 How to work with a requirement block

Each requirement block contains a well-defined requirement. It consists of a defining text, explanatory specifying notes, the documentation to be provided, the validation guidance and examples of acceptable solutions (if available). The content within a requirement block may be subdivided according to risk classes. This leads to the schematic presentation of a requirement block shown in Figure 2-3.

Title of the requirement		
Main statement of the requirement		
Specifying notes (scope of application, additional explanations, exceptional cases, etc.)		
Documentation to be provided (eventually differentiated between risk classes)		
Validation guidance for one risk class	**Validation guidance** for another risk class	**...**
Example of an acceptable solution for one risk class	**Example of an acceptable solution** for another risk class	**...**

Figure 2-3: Structure of a requirement block

The requirement block represents the technical content of the requirement including the validation guidance. It addresses both the manufacturer and the notified body in two directions: (1) to consider the requirement as a minimal condition, and (2) not to put demands beyond this requirement.

Notes for the manufacturer:

- Observe the main statement and the additional specifying notes.
- Provide documentation as required.
- Acceptable solutions are examples that comply with the requirement. There is no obligation to follow them.
- The validation guidance has an informative character.

Notes for notified bodies:

- Observe the main statement and the additional specifying notes.
- Follow the validation guidance.
- Confirm the completeness of the documentation provided.

2.4 How to work with the checklists

Checklists are means of ensuring that all the requirements within a chapter have been covered by the manufacturer or examiner. They are part of the pattern test report. Be aware, the checklists are only of a summarising nature, and they do not distinguish between risk classes. Checklists do not replace the requirement definitions. Refer to the requirement blocks for complete descriptions.

Procedure:

- Gather the checklists, which are necessary according to the selection described in steps 1, 2 and 3 in section 2.2.
- Go through the checklists and prove whether all requirements have been met.
- Fill in the checklists as required.

3 Definition of Risk Classes

3.1 General principle

The specific requirements of this guide are differentiated according to (software) risk classes. In this guide, risks are related to software of the measuring instrument and not to any other component. For convenience reasons, the shorter term “risk class” is used. Each measuring instrument shall be assigned to a risk class because the specific software requirements to be applied are tailored to the risk class the instrument belongs to.

Software risks in measuring instruments addressed by this guide are mainly caused by three risk factors: inadequate protection of software, inadequate examination of software, and non-conformity to type. A risk class is a combination of levels of these three risk factors where the definition of levels of the risk factors is indirectly made by definition of levels for the correspondingly necessary counteractions. Three levels of counteractions, low, middle and high, are introduced for each of the risk factors. The higher the risk is assumed, the higher the level of counteraction is taken.

3.2 Description of levels of counteractions for the risk factors

The following definitions are used for the corresponding levels.

Software protection levels

Low: No particular protection measures against intentional changes are required.

Middle: The software is protected against intentional changes made by using easily-available and simple common software tools (e.g. text editors).

High: The software is protected against intentional changes made by using sophisticated software tools (debuggers and hard disc editors, software development tools, etc).

Software examination levels

Low: Standard type examination including functional testing of the instrument is performed. No extra software testing is required.

Middle: In addition to the low level, the software is examined on the basis of its documentation. The documentation includes the description of the software functions, parameter description, etc. Practical tests of the software-supported functions (spot checks) may be carried out to check the plausibility of documentation and the effectiveness of protection measures.

High: In addition to the middle level, an in-depth test of the software is carried out, usually based on the source code.

Software conformity levels

Low: The legally relevant software of individual instruments is considered conform to the legally relevant software of the type under examination if the functionality of the software corresponds to the technical documentation of the type. The binary code of the software itself may not necessarily be identical to the software of the type.

Middle: In addition to the conformity level “low”, the binary code of legally relevant software of individual instruments is identical to the software of the type under

examination (or re-examination). Software separation is allowed if the restrictions in part S of this guide (chapter 8) are fulfilled.

High: The binary code of the complete software implemented in the individual instruments is identical to the software of the type under examination. Software separation is not anymore relevant.

3.3 Derivation of risk classes

Out of the 27 theoretically possible level combinations, only 3 or at the utmost 6 are of practical interest (risk classes B, C, D and eventually A, E and F). They cover all of the instrument classes falling under the regulation of MID. Moreover, they provide a sufficient window of opportunity for the case of changing risk evaluations. The classes are defined in the table below. The table shall be interpreted in a way that a certain risk class is defined by the corresponding combination of levels of necessary counteractions.

Risk Class	Software Protection	Software Examination	Software Conformity
A	*low*	*Low*	*low*
B	*middle*	*Middle*	*low*
C	*middle*	*Middle*	*middle*
D	*high*	*Middle*	*middle*
E	*high*	*High*	*middle*
F	*high*	*High*	*high*

Table 3-1: *Definition of risk classes*

3.4 Interpretation of risk classes

Risk class A: It is the lowest risk class at all. No particular measures are required against intentional changes of software. Examination of software is part of the functional testing of the device. Conformity is required on the level of documentation. It is not expected that any instrument is classified as a risk class A instrument. However, by introducing this class, the corresponding possibility is held open.

Risk class B: In comparison to risk class A, the protection of software is required on the middle level. Correspondingly, the examination level is raised to the middle level. The conformity remains unchanged in comparison to risk class A.

The software examination is carried out on the basis of the documentation. In the consequence, the TEC allows different implementations with respect to the same documentation when putting the instruments into market[1].

Risk class C: In comparison to risk class B, the conformity level is raised to “middle”. This means, the binary code of the legally relevant software of individual instruments is identical to the software of the type under examination. The levels of protection and examination remain unchanged in comparison to risk class B.

[1] After having put the instrument into market, the allowance for changing software depends on national regulations.

Risk class D: The significant difference in comparison to risk class C is the upgrade of the protection level to “high”. The examination level remains unaffected at “middle”, therefore sufficiently informative documentation shall be provided to show that the protection measures taken are appropriate. The conformity level remains unchanged in comparison to risk class C.

Risk class E: In comparison to risk class D, the examination level is raised to “high”. The levels of protection and conformity remain unchanged.

Risk class F: The levels with respect to all aspects (protection, examination and conformity) are set to “high”. The difference to risk class E is that there is not any legally non-relevant software anymore.

4 Basic Requirements for Embedded Software in a Built-for-purpose Measuring Instrument (Type P)

The set of specific requirements of this chapter are valid for built-for-purpose instruments as well as for sub-assemblies and for parts according to WELMEC Guide 8.8 (Modular Evaluation of Measuring instruments) that are of the built-for-purpose type. The validity for sub-assemblies and parts is included even if it is not repeatedly mentioned in the following text. The conditions, however, under which sub-assemblies and parts may be separately examined and the corresponding certificates may be accepted, are not part of this guide.
If the measuring instrument uses a universal computer (general purpose PC), the set of specific requirements in chapter 5 shall be referred to (Type U instrument). The specific requirements of type U instruments shall always be used if at least one of the subsequent technical characteristics of built-for-purpose instruments is not matched.

4.1 Technical Description

A type P instrument is a measuring instrument with an embedded IT system (e.g., a microprocessor or microcontroller based system). *All components of the IT system used are open for evaluation.*

The embedded IT system is in particular characterised as follows:

- The software is exclusively constructed for the measuring purpose. Additional functions for securing software and data, for transmitting data and for downloading software are considered constructed for the measuring purpose.
- The user interface is dedicated to the measuring purpose, i.e. it is normally in an operating mode subject to legal control. Switching to an operating mode not subject to legal control is possible.
- An operating system (OS) or subsystems of it may be included if
  - all communication is under control of legally relevant software,
  - it does not allow loading or changing programs, parameters or data or running programs,
  - if it does not allows to change the environment of the legally relevant application, etc.

  This includes that the access prevention shall be preset and not the result of a respective subsequent configuration of these components.
- The software environment is invariable and there are no internal or external means for programming or changing the software in its embedded status. Software download is allowed if the specific requirements of extension D (chapter 9) are observed.

4.2 Specific Requirements for Type P

Risk Classes B to E
P1: Documentation *In addition to the specific documentation required in each of the following requirements, the documentation shall basically include:* *a. A description of the legally relevant software.* *b. A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).* *c. A description of the user interface, menus and dialogues.* *d. The software identifier(s) of the legally relevant software.* *e. An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc.* *f. The operating manual.*

Risk Class B	Risk Class C	Risk Class D
P2: Software identification *The legally relevant software shall be clearly identified. The identifier(s) shall be permanently presented by the instrument or presented on command or during operation.*		
Specifying Notes: 1. Legally relevant software identifiers may be independent or part of well structured identifiers. In the second case, the legally relevant software identifier(s) shall be clearly distinguishable. 2. If different software versions are valid implementations of the same type (e.g., for instruments in risk class B), then the legally relevant software identifier(s) shall be unique for each version 3. The legally relevant software identifiers are considered to be type-specific parameters. If the identifier is inextricably linked to the software itself, the securing means for software apply (see P5 and P6). If not, other securing means are required. 4. The legally relevant software identifiers shall be easily presented without requiring an additional tool.		
Required Documentation: 1. The documentation shall list the software identifier(s) and describe how they are created, how they are secured, how they are presented and how they are structured in order to differentiate between legally relevant software identifiers and others as well as to assess the uniqueness.		
Validation Guidance: *Checks based on documentation:* • Check whether legally relevant software identifiers are given in the documentation. • Check whether the software performing legally relevant functions is clearly described so that it is reproducible which legally relevant software part is covered by which legally relevant software identifier. • Examine the description of the visualisation of the legally relevant software identifiers. • Check whether all legally relevant software identifiers are unique (in particular in cases of re-examinations). *Functional Checks:* • Check that the legally relevant software identifiers can be visualised as described in the documentation. • Check that the legally relevant software identifier(s) presented are identical to the identifiers given in the documentation. • The legally relevant software identifier(s) are distinguishable from other identifiers.		
Example of an Acceptable Solution: • Several formats of the legally relevant software identifier are acceptable: a) any string of numbers, letters, other characters, b) any string, possibly added by a version number, c) a checksum over code. • If the manufacturer chooses a mixed identifier for legally relevant and legally non-relevant software, a simple solution that allows distinguishing the identifiers is using placeholders in the TEC, e.g. “abc1.xx” with “abc1” for the legally relevant software and “xx” as placeholder for legally non-relevant software. The identifier(s) are displayed permanently on a secured plate, on command or on start-up.		

Additions for Risk Class E
Required Documentation Identical to risk classes B to D.
Validation Guidance Identical to risk classes B to D.

Risk Class B	Risk Class C	Risk Class D

P3: Influence via user interface

Commands entered via the user interface shall not inadmissibly influence the legally relevant software, device-specific parameters *and measurement data*.

Specifying Notes:

1. There shall be an unambiguous assignment of each command to an initiated function or data change.
2. Commands that are not documented shall have no effect on legally relevant functions, device-specific parameters and measurement data.
3. The respective parts of the software that interpret commands are considered to be legally relevant software.

Required Documentation:

If the instrument has the ability to receive commands, the documentation shall include:

- Description of commands and their effect on legally relevant software, device-specific parameters and measurement data.
- Description of how the legally relevant software, device-specific parameters and measurement data are protected from being influenced by other inputs.

Validation Guidance:

Checks based on documentation:

- Check that documented commands are admissible, i.e., that they have an allowed influence on the legally relevant software, device-specific parameters and measurement data).
- Check the protection measures against influences from other inputs.

Functional Checks:

- Carry out practical tests (spot checks) with documented commands.
- Check whether there are undocumented commands.

Example of an Acceptable Solution:

There is a software module that receives and interprets commands from the user interface. This module belongs to the legally relevant software. It forwards only allowed commands to the other legally relevant software modules. All unknown or not allowed sequences of switch or key actuations are rejected and have no impact on the legally relevant software, device-specific parameters and measurement data.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B to D)**:**

Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes B to D)**:**

Checks based on the source code:

- Check the software design whether data flow concerning commands is unambiguously defined and realised only in the legally relevant software.
- Search inadmissible data flow from the user interface to domains to be protected.
- Check with tools or manually that commands are decoded correctly.

Risk Class B	Risk Class C	Risk Class D

P4: Influence via communication interface

Commands inputted via communication interfaces of the instrument shall not inadmissibly influence the legally relevant software, device-specific parameters *and measurement data.*

Specifying Notes:

1. There shall be an unambiguous assignment of each command to an initiated function or data change.
2. Commands that are not documented shall have no effect on legally relevant functions, device-specific parameters and measurement data.
3. The respective parts of the software that interpret commands are considered to be legally relevant software.
4. Interfaces that allow commands with inadmissible effects on the legally relevant software, device-specific parameters and measurement data shall be sealed or protected in another appropriate way. This also applies for interfaces that cannot be completely assessed.
5. This special requirement does not apply to software download according to Extension D.

Required Documentation:

If the instrument has an interface, the documentation shall include:

- Description of commands and their effect on the legally relevant software, device-specific parameters and measurement data.
- Description of how the legally relevant software, device-specific parameters and measurement data are protected from being influenced by other inputs.

Validation Guidance:

Checks based on documentation:

- Check–that documented commands are admissible, i.e., that they have an allowed influence on the legally relevant software, device-specific parameters and measurement data).
- Check the protection measures against influences from other inputs.

Functional checks:

Carry out practical tests (spot checks) using peripheral equipment.

Example of an Acceptable Solution:

There is a software module that receives and interprets data from the interface. This module is part of the legally relevant software. It forwards only allowed commands to the other legally relevant software modules. All unknown or not allowed signal or code sequences are rejected and have no impact on the legally relevant software, device-specific parameters and measurement data.

Additions for Risk Classes E

Required Documentation (in addition to the documentation required for risk classes B to D)**:**

Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes B to D)**:**

Checks based on the source code:

- Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified.
- Search inadmissible data flow from the interface to domains to be protected.
- Check with tools or manually that commands are decoded correctly.

Risk Class B	Risk Class C	Risk Class D

P5: Protection against accidental or unintentional changes

Legally relevant software and device-specific parameters shall be protected against accidental or unintentional changes.

Specifying Notes:

1. The software shall be capable to detect changes caused by physical effects (electromagnetic interference, temperature, vibration, etc).
2. Means shall be implemented to protect from unintentional misuse of the user interfaces.

Required Documentation:

1. The documentation should show the measures that have been taken to detect and protect the legally relevant software and device-specific parameters from unintentional changes.

Validation Guidance:

Checks based on documentation:

- Check that measures against unintentional changes are described and appropriate.

Functional checks:

- Practical spot checks to show that a warning is given before deleting measurement data, if deleting is possible at all.

Example of an Acceptable Solution:

- The accidental modification of legally relevant software and device-specific parameters shall be checked by periodically calculating checksum(s) and automatically comparing them with deposited nominal value(s). If the comparison does not match, reactions are necessary that are adequate for the instrument (e.g., stop of measurement, corresponding indication of measurement data, see chapter 10 for eventual recommendations).
- Alternative methods are possible if the change status of software can be identified by them.
- For fault detection see Extension I (chapter 10).

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes C and D):
Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes C and D)**:**

Checks based on the source code:

- Check whether measures taken for detection of changes are appropriate.
- Check whether all parts of the legally relevant software are covered by the checksum.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">P6: Protection against intentional changes
Legally relevant software and measurement data shall be secured against inadmissible modification, loading or swapping of hardware memory.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. For protection against manipulation using the user interface, see P3.
2. For protection against manipulation using communication interfaces, see P4.
3. Measurement data are already considered to be sufficiently protected, if only legally relevant software processes them (e.g. in memory or registers).</td></tr>
<tr><td colspan="2"></td><td>Specifying Notes:
4. A checksum or an alternative method with the same level of requirements shall be provided in order to support the detection of software modifications.
5. The calculated checksum or an alternative indication of software modification shall be made visible on command for control purposes.
6. The checksum or the alternative indication is calculated over the legally relevant software. The software that organizes the generation of checksums or alternative indications is part of the legally relevant software.</td></tr>
<tr><td colspan="3">Required Documentation:
The documentation shall describe the protection methods.</td></tr>
<tr><td colspan="3">• Description of measures that have been taken to protect the software and device-specific parameters, in particular for Risk Class C and D the method of checksum calculation and nominal checksums with the corresponding nominal indication.</td></tr>
<tr><td colspan="3">Validation Guidance:
Checks based on documentation:
• Examine whether the documented means of securing against unauthorised exchange of the memory that contains the software are sufficient.
• Check that the checksum(s) or alternative indication(s) cover the legally relevant software.
Functional checks:
• Test practically the programming mode and check whether disabling works.
• Compare calculated checksums or alternative indications with the nominal values.</td></tr>
<tr><td colspan="2">Example of an acceptable Solution:
a) To prevent from removing and replacing physical memory, the housing of the instrument or the physical memory itself shall be secured against unauthorised removal.
b) The instrument is sealed and the interfaces comply with the requirements P3 and P4.</td><td>Example of an acceptable Solution: (in addition to a) and b))
c) Program code is protected by means of checksums. The program calculates its own checksum and compares it with a desired value that is hidden in the executable code. If the self-check fails, the program is blocked.
Any checksum algorithm should have a key length of at least 2 bytes; a CRC-32 checksum with a secret initial vector (hidden in the executable code) would be satisfactory. (See also Extensions L and T).</td></tr>
</table>

Additions for Risk Classes E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for the detection of intentional changes are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">P7: Parameter protection
Device-specific parameters shall be secured against unauthorised modification after setting.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. In normal secured operating mode device specific parameters shall not be alterable any more. They shall only be adjustable in a special operating mode of the instrument.
2. There may be device-specific parameters that are allowed to remain unsecured. See extension I for instrument specific parameters.</td></tr>
<tr><td colspan="3">Required Documentation:
The documentation shall describe the device-specific parameters, whether they may be set and how they are set and how they are secured.</td></tr>
<tr><td colspan="3">Validation Guidance:
Checks based on documentation:
• Check that changing or adjusting of device specific parameters is impossible after securing.
• Check that all relevant parameters (given in Extension I, if any) are secured.
Functional checks:
• Test the adjusting (configuration) mode and check whether disabling after securing works.
• Examine the classification and state of parameters (secured/settable) at the display of the instrument, if a suitable menu item is provided.</td></tr>
<tr><td colspan="3">Example of an Acceptable Solution:
a) Parameters are secured by sealing the instrument or memory housing and disabling the write enable/disable input of the memory circuit by an associated jumper or switch, which is sealed.</td></tr>
<tr><td colspan="2">b) Event counter / event logger:
• An event counter registers each change of a parameter value. The current count can be displayed and can be compared with the initial value of the counter that was registered before putting the measuring instrument into use or at the last official verification respectively and is indelibly labelled on the instrument.
• Changes of parameters are registered in an event logger. It is an information record stored in a non-volatile memory. Each entry is generated automatically by the legally relevant software and contains:
○ the identifier of the parameter (e.g. the name)
○ the parameter value (the current or the value before)
○ the time stamp of the change
• The event logger cannot be deleted or be changed without destroying a seal.

<u>Defining note:</u>
Event counter: An event counter registers each change of a parameter value. It serves as a means to supervise changes.
Event logger: An event logger registers each change of software or parameters. It serves as a means to supervise changes. It registers at least the identifier of the changed item, and additional information.</td><td>÷</td></tr>
</table>

Additions for Risk Classes E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software showing the way of securing and viewing legally relevant parameters.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check code whether measures taken for protecting parameters are appropriate (e.g. adjusting mode disabled after securing).

5 Basic Requirements for Software of Measuring Instruments using a Universal Computer (Type U)

The set of specific requirements of this chapter is valid for measuring instruments based on a general-purpose computer as well as for sub-assemblies and for parts according to WELMEC guide 8.8 that uses universal computers. The validity for sub-assemblies and parts is included even if it is not repeatedly mentioned in the following text. The conditions, however, under which sub-assemblies and parts may be separately examined and the corresponding certificates may be accepted, are not part of this guide.

5.1 Technical Description

A type U measuring system is typically characterised by the following configurations.

Hardware Configuration

a) A modular general-purpose computer-based system. The computer system may be stand-alone, part of a closed network, e.g. Ethernet, token-ring LAN, or part of an open network, e.g. Internet.

b) Because the system is general purpose, the sensor is normally external to the computer unit and linked to it by a communication connection.

c) The user interface offers further functions, which are not under legal control, besides the operating mode for the measurement task.

d) Storage may be fixed, e.g. hard disk, removable, e.g. USB, or remote.

Software Configuration

e) Usually, an operating system is used.

f) In addition to the measuring instrument application, other software applications may also reside on the system at the same time.

In addition to configurations described above, a type U system shall also be assumed if the characteristics of a type P instrument (see Chapter 4.1) are not completely fulfilled.

Off-the-shelf operating system and low level drivers supplied together with them, e.g. video drivers, printer drivers, disk drivers, etc., are not considered as legally relevant unless parts are replaced by alternative ones or specially programmed for a specific measuring task.

Consequences for risk classification

The software of type U instruments is much more openly accessible than the software of type P instruments. The protection of software integrity shall be enhanced in comparison to type P instruments. In particular, a checksum or an equivalent means shall be required to support integrity checks of the software code. The consequence is that the conformity level “low” (only functional correspondence of the software to the technical documentation of the type under examination) is not an adequate means for ensuring software integrity. This means risk class C is the lowest possible risk class instruments of the U type may be allocated to.

5.2 Specific Software Requirements for Type U

Risk Classes C to E

U1: Documentation

In addition to the specific documentation required in each requirement below, the documentation shall basically include:

a. *A description of the legally relevant software functions, meaning of the data, etc.*
b. *A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).*
c. *A description of the user interface, menus and dialogues.*
d. *An identifier of the legally relevant software .*
e. *An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc.*
f. *An overview of the configuration of the operating system used, security aspects of the operating system utilised, e.g. protection, user accounts, privileges, etc.*
g. *The operating manual.*

Risk Class C and D

U2: Software identification

The legally relevant software shall be clearly identified. The identifier(s) *shall be permanently presented by the instrument, presented on command or during operation.*

Specifying Notes:

1. Legally relevant software identifier(s) may be independent or part of well structured identifiers.
2. In the case that a legally relevant software identifier is embedded in an overall identifier, it shall be clearly distinguishable.
3. The legally relevant identifier(s) shall be unique for each the legally relevant software an instrument is equipped with.
4. The legally relevant identifiers shall be easily presented without requiring an additional tool.
5. Identification shall include drivers and components of operating systems that have been modified or specifically programmed for a legally relevant task. Standard components used unchanged may be excluded from identification.
6. If the legally relevant functions and the account of the measuring task are protected by a specific configuration of the operating system, the relevant configuration files shall have an own identifier.
7. The legally relevant software identifier(s) are considered to be type-specific parameters and shall be protected as such (see U5 and U6). If the identifiers are not inextricably linked to the software itself, other securing means are required.

Required Documentation:

The documentation shall list the software identifiers and describe how they are created, how they are secured, how they are presented and, if applicable, how they are structured in order to differentiate between legally relevant identifiers and others.

Validation Guidance:

Checks based on documentation:

- Check whether legally relevant software identifiers are given in the documentation.
- Check whether the software performing the legally relevant tasks is clearly described so that it is reproducible which software part is covered by which software identifier.
- Examine the description of generation and visualisation of identifiers.
- Check whether there are modified or self-developed components of an operating system and, if yes, whether they are included in identification.
- If the software for measuring functions is protected by a specific configuration of the operating system, check whether the relevant configuration file(s) have own identifier(s).
- Check whether all legally relevant software identifiers are unique.

Functional checks:

- The software identifiers can be visualised as described in the documentation.
- The presented identifiers are identical to the identifiers given in the documentation.
- The legally relevant identifiers are distinguishable from other identifiers.

Example of an Acceptable Solution:

- Several formats of legally relevant software identifiers are acceptable:
  - a) any string of numbers, letters, other characters,
  - b) a string added by a version number,
  - c) a checksum over code.
- If the manufacturer chooses a mixed identifier for legally relevant and legally non-relevant software, a simple solution that allows distinguishing the identifiers is using placeholders in the TEC, e.g. “abc1.xx” with “abc1” for the legally relevant software and “xx” as placeholder for legally non-relevant software.
- The identifier(s) are displayed permanently, on command or on start-up.

Additions for Risk Class E
Required Documentation Identical to risk classes C and D.
Validation Guidance Identical to risk classes C and D.

Risk Class C	**Risk Class D**
U3: Influence via user interfaces *Commands entered via the user interface shall not inadmissibly influence legally relevant software, device-specific parameters and measurement data.*	
Specifying Notes: 1. There shall be an unambiguous assignment of each command to an initiated function or data change. 2. Commands that are not documented shall have no effect on legally relevant functions, device-specific parameters and measurement data. 3. The respective parts of the software that interpret commands are considered to be legally relevant software. 4. In particular, functions of the operating system offered at the user interface shall not influence the legally relevant software, device-specific parameters and measurement data including the configuration of the operating system or other means for their protection.	
÷	5. The user shell shall be closed, i.e. the user shall not be able to load programs, write programs or perform commands to the operating system.
Required Documentation: If the instrument has the ability to receive commands, the documentation shall include: • Description of commands and their effect on legally relevant software, device-specific parameters and measurement data. • Description of how the legally relevant software, device-specific parameters and measurement data are protected from being influenced by other inputs. • In particular, description of how the legally relevant software, device-specific parameters and measurement data are protected from functions of the operating system offered to the user.	**Required Documentation** (in addition to the documentation required for risk class C)**:** • Description of protections means against other inputs including functions of the operating system offered to the user.
Validation Guidance: *Checks based on documentation:* • Check–that documented commands are admissible, i.e. that they have an allowed influence on the legally relevant software, device-specific parameters and measurement data). • Check the protection measures against influences from other commands. • In particular, check the protection measures against influences from functions of the operating system offered to the user. *Functional checks:* • Carry out practical tests (spot checks) with documented commands. • Check whether there are undocumented commands. .	**Validation Guidance** (in addition to the guidance for risk class C)**:** *Checks based on documentation:* • Check whether the measures taken and test protocols are appropriate for the high protection level.

Example of acceptable Solution: • A module in the legally relevant software filters out inadmissible commands. Only this module receives commands, and there is no circumvention of it. Any false input is blocked.	**Example of acceptable Solution:** • For using the measuring system, only an account with restricted permissions is set up. Access to the administrator account is blocked according to U6.

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk class D)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk class D)**:** *Checks based on the source code:* • Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified. • Search inadmissible data flow from the user interface to domains to be protected. • Check with tools or manually that commands are decoded correctly.

Risk Class C	Risk Class D

U4: Influence via communication interface

Commands input via communication interfaces of the device shall not inadmissibly influence the legally relevant software, device-specific parameters *and measurement data.*

Specifying Notes:

1. There shall be an unambiguous assignment of each command to an initiated function or data change.
2. Commands that are not documented shall not have any effect on legally relevant functions, device-specific parameters and measurement data.
3. The respective parts of the software that interpret commands are considered to be legally relevant software.
4. Interfaces that allow commands with inadmissible effects on the legally relevant software, device-specific parameters and measurement data shall be sealed or protected in another appropriate way. This also applies for interfaces that cannot be completely assessed.
5. This special requirement does not apply to software download according to Extension D.

Please note: If the operating system allows remote control or remote access, the requirements U3 apply to the communication interface and the connected remote terminal, respectively.

Required Documentation:

If the instrument has an interface, the documentation shall include:

- Description of commands and their effect on legally relevant software, device-specific parameters and measurement data.
- Description of how the legally relevant software, device-specific parameters and measurement data are protected from being influenced by other inputs.

Validation Guidance:

Checks based on documentation:

- Check that documented commands are admissible, i.e. that they have an allowed influence on the legally relevant software, device-specific parameters and measurement data).
- Check the protection measures against influences from other commands.

Functional checks:

- Carry out practical tests (spot checks) using peripheral equipment.

Example of an Acceptable Solution:

There is a software module that receives and interprets commands from the interface. This module belongs to the legally relevant software. It forwards only allowed commands to the other legally relevant software modules. All unknown or not allowed commands are rejected and have no impact on the legally relevant software, device-specific parameters and measurement data.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes C to D)**:**
Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes C to D)**:**

Checks based on the source code:

- Check the software design whether data flow concerning commands is unambiguously defined in the legally relevant software and can be verified.
- Search inadmissible data flow from the interface to domains to be protected.
- Check with tools or manually that commands are decoded correctly.

Risk Class C	Risk Class D

U5: Protection against accidental or unintentional changes

Legally relevant software and device specific parameters shall be protected against accidental or unintentional changes.

Specifying Notes:

1. The software shall be capable to detect changes caused by physical effects (electromagnetic interference, temperature, vibration, etc).
2. Means shall be implemented to protect from unintentional misuse of the user interfaces.
3. The accidental modification of legally relevant software and device-specific parameters shall be periodically checked by calculating checksum(s) and automatically comparing them with deposited nominal value(s). If the comparison does not match, reactions are necessary that are adequate for the instrument (stop of measurement, indication of measurement data, see chapter 10 for eventual recommendations) Alternative method are possible if the change status of software can be identified by them. .

Required Documentation:

- Description of measures that have been taken to detect and protect the legally relevant software and device-specific parameters from unintentional changes.
- Description of the checksum method and of reactions in case of non-matching.
- Description of how and where the nominal checksum(s), or the alternative indications of change status, are deposited.

Validation Guidance:

Checks based on documentation:

- Check that measures against unintentional changes are described and appropriate.
- Check that the checksum(s) comprise the legally relevant software.
- Check that methods of checksum calculation, comparison and of reactions in the case of non-matching are correct.

Example of an Acceptable Solution:

- Misuse of the operating system, overwriting or deletion of stored data and programs: It is made full use of the protection or privacy rights provided by the operating system or programming language.
- The accidental modification of legally relevant software is checked by calculating a checksum over the relevant code, comparing it with the nominal value and initiating appropriate actions if the code has been modified.
- Where the operating system allows it, it is recommended that all user rights for the deletion, moving or amendment of legally relevant software is removed and access is controlled via utility programs.
- Access control to legally relevant software by the use of passwords is recommended, as is the use of read-only mechanisms. The system supervisor should restore rights only when required.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes C to D)**:**
Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes C to D)**:**

Checks based on the source code:

- Check whether measures taken for detection of changes (faults) are appropriate.
- Check whether all parts of the legally relevant software are covered by the checksum.

<table>
<tr><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="2">U6: Protection against intentional changes
Legally relevant software and measurement data shall be secured against intended, inadmissible modification or replacement.</td></tr>
<tr><td colspan="2">Specifying Notes:
1. Mass storage device where legally relevant software, configuration files and device-specific parameters are stored shall be protected against physical exchange.
2. A checksum or an alternative method with the same level of requirements shall be provided in order to support the detection of software modifications. The calculated checksum or an alternative indication of software modification shall be made visible on command for control purposes.
3. The checksum or the alternative indication is calculated over the legally relevant software. The software that organizes the generation of checksums or alternative indications is part of the legally relevant software.
4. Measures shall be taken to protect legally relevant software from being modified or replaced by other software using the protection means of the operating system.
5. The parts and features of the operating system that implement the protection of legally relevant software shall be also considered as legally relevant software and be protected as such.
6. This special requirement does not apply to software download according to extension D.</td></tr>
<tr><td></td><td>7. In general, a universal computer is only usable if additional hardware can be used to support securing.</td></tr>
<tr><td colspan="2">Required Documentation:
• Description of measures that have been taken to protect the software and device-specific parameters, in particular method of checksum calculation and nominal checksums or alternative method with the corresponding nominal indication.
• Description of methods how the mass storages are protected from exchange, if applicable.
• Description of used securing features of operating system.
• Description of how the checksum or an alternative indication are presented.</td></tr>
<tr><td colspan="2">Validation Guidance:
Checks based on documentation
• Check that the checksum(s) or alternative indication(s) comprise the legally relevant software.
• Check that measures taken to prevent from modifying or replacing legally relevant software by using the operation system are adequate.
• Check that features of the operating system used for the protection of legally relevant software are part of legally relevant software and secured as such.
• Check that mass storage devices are protected from being physically exchanged, if applicable.

Functional checks
• Arrange to be calculated checksums or alternative indications and compare with the nominal values.</td></tr>
<tr><td></td><td>Checks based on documentation:
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td>Example of an Acceptable Solution:
• Program code is protected by means of checksums. The program is calculating its own checksum and compares it with a desired value that is hidden in the executable code. If the self-check fails, the program is blocked.
• Any checksum algorithm should have a key length of at least 2 bytes; a CRC-32 checksum with a secret initial vector (hidden in the executable code) would be satisfactory. (See also Extensions L and T).
• The unauthorised manipulation of legally relevant software may be inhibited by the access control or privacy protection attributes of the operating system. The administration level of these systems shall be</td><td>Example of an Acceptable Solution:
• Program code may be secured by storing the legally relevant software in a dedicated plug-in-unit, which is sealed. The plug-in unit may include, for example, a read-only memory and a microcontroller.
• Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).</td></tr>
</table>

secured by sealing or equivalent means. • The access to the administrator account is a) blocked for everyone or b) only granted to authorised persons as regulated by the national market surveillance laws. Solution a) Random password generated automatically, known to nobody. Change of the legally relevant configuration only possible by performing a new operating system set up. Solution b) Password chosen by the authorised person and hidden and sealed in a envelope or in /at the housing. • Circumvention of the protection means of the operating system by direct writing to mass storages or physical replacement is prohibited by sealing.	

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes C to D)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance required for risk class D)**:** *Checks based on the source code:* • Check communication with the additional securing hardware. • Check that changes of legally relevant software are detected.

Risk Class C	**Risk Class D**
U7: Parameter protection *Device-specific parameters shall be secured against unauthorised modification after setting.*	
Specifying Notes: 1. Because settable device specific parameters could be manipulated using simple tools on universal computers, they shall be stored in secured hardware, e.g. in the respective sensor.	
Required Documentation: The documentation shall describe the device-specific parameters, whether they may be set and how they are set and how they are secured.	
Validation Guidance: *Checks based on documentation:* • Check that changing or adjusting of device specific parameters is impossible after setting. • Check that all relevant parameters are secured.	
Example of an Acceptable Solution: • Device specific parameters to be protected are stored on a plugged-in storage which is sealed against removing or directly on the sensor unit. Writing of parameters is inhibited by sealing a write-enable switch in the disabled state. • Unprotected settable parameters are stored on a standard storage of the universal computer.	

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes C to D)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes C to D)**:** *Checks based on the source code:* • Check whether measures taken for protecting parameters are appropriate.

Risk Class C	**Risk Class D**
U8: Presentation of measurement data *The authenticity of the measurement data that are presented shall be guaranteed.*	

Specifying Notes:

1. Presented measurement data are considered authentic if the presentation is issued from within the legally relevant software.
2. It shall not be possible to fraudulently simulate (spoof) legally relevant software for presenting measurement data using the capabilities of the operating system or other easily available and manageable tools.
3. Presented measurement data shall be comprehensible and clearly distinguishable from other, legally non-relevant information. If necessary, additional explanation shall be given.
4. If it is not possible to realise full protection by the capabilities of the operating system, it shall be ensured by technical means that on the universal computer only the legally relevant software can per form the legally relevant functions (e.g. a sensor shall only work together with the legally relevant indicating program on the universal computer).

Required Documentation:

The documentation should describe how authenticity of the measurement data is guaranteed.

Validation Guidance:

Checks based on documentation:

- Check that presented measurement data is generated by legally relevant software.
- Check that the presentation of measurement data can only be performed by legally relevant software.

Functional checks:

- Check through visual control if the presentation of measurement data is easily distinguishable from other information that may also be presented.

Examples of an Acceptable Solution:

1. A measurement application window is generated by the legally relevant software. The technical measures required of the window are:
   - No access to measurement data shall be given to legally non-relevant programs until the measurement data have been indicated.
   - The window is refreshed periodically. The associated program checks that it is on top of the stack of windows and the user shall not be enabled to close the window or shift it outside the visible area as long as the measurement is not concluded.
   - Processing of measurement values stops whenever this window is closed or not completely visible.

2a The sensor unit encrypts the measuring values with a key known to the authentic software running on the universal computer (e.g. its version number). Only the authentic software can decrypt and use the measurement values, non-authentic programs on the universal computer cannot as they do not know the key. For key treatment see Extension T.

2b Before sending measurement values the sensor initiates a handshake sequence with the legally relevant software on the universal computer based on secret keys. Only if the program on the universal computer communicates correctly, the sensor unit sends its measurement values. For key treatment see Extension T.

3. The key used in 2a / 2b may be chosen and entered to the sensor unit and software on the universal computer without destroying a seal.	3. The key used in 2a / 2b is the hash code of the program on the universal computer. Each time the software on the universal computer is changed; the new key is entered into the sensor unit and is secured in a way that the seal must be broken to change it.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes C to D)**:**
Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes C to D)**:**

Checks based on the source code:

- Check that legally relevant software generates the presented measurement data.
- Check whether all measures taken are appropriate and correct to guarantee the presentation of measurement results by legally relevant software.

Risk Classes C to E

U9: Influence of other software

The legally relevant software shall be designed in such a way that other software does not inadmissibly influence it.

Specifying Notes: 1. This requirement implies software separation between the legally relevant and legally non-relevant software under consideration of the state-of-the-art of software engineering for modularisation or object oriented concepts. Extension S shall be observed. This is the standard case for universal computers.
Required Documentation: See Extension S.
Validation Guidance: See Extension S.
Example of an Acceptable Solution: See Extension S.

6 Extension L: Long-term Storage of Measurement Data

The specific requirements of this section only apply if long-term storage of measurement data is designed. They are an addition to the specific requirements of embedded software for built-for-purpose measuring instrument (type P requirements) and of software for measuring instruments using a universal computer (type U requirements).

Long-term storage includes the time from when a measurement is physically completed to the point in time when all processes to be done by the *legally relevant software* are finished. It may also be applied to long-term storage of the data thereafter.

6.1 Technical description

Three different technical configurations for long-term storage are listed in the following table. For a built-for-purpose device, the variant of an integrated storage is typical: here the storage is part of the metrologically necessary hardware and software. For instruments using a universal computer, another variant is typical: the use of resources already existing, e.g., hard disks. The third variant is the removable storage: here the storage can be removed from the device, which could be either a built-for-purpose device or a universal computer, and be taken elsewhere. When data is retrieved from removable storage for legal purposes, e.g. visualisation, ticket printing, etc, the retrieving device shall be subject to legal control.

A) Integrated storage
Simple instrument, built-for-purpose, no externally usable tools or means available for editing or changing data, integrated storage for measurement data or parameters, e.g. RAM, flash memory, hard disk.
B) Storage for universal computer
Universal computer, graphical user interface, multitasking operating system, tasks subject to legal control and not subject to legal control exist in parallel, storage can be removed from the device or contents can be copied anywhere inside or outside the computer.
C) Removable or remote (external) storage
Arbitrary basic instrument (built-for-purpose instrument or instrument using universal computer), storage can be taken from the instrument. These can be, for example, USB stick, flash cards, or remote databases connected via network.

Table 6-1: Technical description of long-term storages

The classification may be reduced for selected kinds of measuring instruments on conclusion of the responsible WELMEC Working Groups, see section 10.

6.2 Specific software requirements for Long-term Storage

Risk Class B	Risk Class C	Risk Class D
L1 Completeness of measurement data stored *The measurement data stored shall be accompanied by all relevant information needed for legally relevant purposes.* **Specifying Notes:** 1. The measurement data stored shall be capable of being traced back to the measurement that has generated the data. 2. The measurement data stored shall be sufficient for checking invoices. 3. The kind of necessary information may depend on the type of instrument. 4. A presupposition to comply with this special requirement is an identification of each data set stored.		
Required Documentation: Description of all fields of the data sets.		
Validation Guidance: *Checks based on documentation:* • Check whether all information needed for legally relevant purposes are contained in the data set.		
Example of an Acceptable Solution: • A legally and metrologically complete data set comprises the following fields: º Measurement value(s) with correct resolution º the unit of measure º the unit price or the price to pay (if applicable) º the date and time of the measurement (if applicable) º identifier of the instrument º the place of the measurement (if applicable) • Data is stored with the same resolution, values, units etc as indicated or printed on a delivery note.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D): Source code of the legally relevant software that generates the data sets for storing.
Validation Guidance (in addition to the guidance for risk classes B to D): *Checks based on the source code:* • Check whether the data sets are correctly built.

Risk Class B	Risk Class C	Risk Class D
L2: Protection against accidental or unintentional changes *Stored measurement data shall be protected against accidental and unintentional changes.* **Specifying Notes:** 1. Data stored shall be capable to detect accidental data changes caused by physical effects (electromagnetic interference, temperature, vibration, etc). 2. Means shall be implemented to protect from unintentional change or deletion of measurement data.		
Required Documentation: Description of protection measures.		
Validation Guidance: *Checks based on documentation:* • Check that a method is implemented to detect accidental data changes. • Check that the method captures all data. . • Check that overwriting of data cannot occur before the end of the data storage period that is foreseen. • Check that a warning is issued to the user if he is about to change or delete measurement data files. *Functional checks:* • Check by practical spot checks that before changing/deleting measurement data a warning is given, if changing/deleting is possible at all.		

Example of an Acceptable Solution:

- Stored measurement data shall be accompanied by additional redundant information to enable the software retrieving, evaluating, and indicating of the data (see L6)
- To detect data changes due to physical effects, a checksum with at least the **CRC-16** algorithm is calculated over the entire data set and inserted into the data set to be stored.

  Note: The algorithm is not secret and, in contrast to requirement L3, neither is the initial vector of the CRC-register nor the generator polynomial i.e. the divisor in the algorithm. The initial vector and generator polynomial are known to both of the programs that create and verify the checksums.
- Measurement data/invoice files are protected by attaching an automatic date stamp on creation and a flag or label stating whether invoices were paid/unpaid. A utility program would only move/delete files if invoices had been paid or were out-of-date.
- Measurement data is not deleted without prior authorisation, e.g. a dialogue statement or window asking for confirmation of deletion.
- Automatic overwriting of measurement data can be allowed if there is adequate protection of the records to be retained. A parameter determining the number of days before measurement data can be deleted may be set and secured when putting into use according to the user's needs and data storage size. The instrument shall stop if the memory is full and all the records are not old enough to be overwritten. Manual deletion (with prior authorisation) may be performed in that case. In cases where an interruption of measurement is problematic (e.g. utility meters), the storage size shall be sufficient to avoid an interruption due to insufficient memory space.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B to D)**:**
Source code of the legally relevant software that realises the protection of stored data.

Validation Guidance (in addition to the guidance for risk classes B to D)**:**

Checks based on the source code:

- Check whether measures taken for protecting stored date are appropriate and correctly implemented.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">L3: Integrity of data
The measurement data stored shall be protected against intentional changes.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. Stored data in integrated storages in general are protected by hardware means. No extra software protection is necessary.
2. The protection shall apply against intentional changes carried out by easily available and manageable software tools.</td></tr>
<tr><td colspan="2"></td><td>3. The protection shall also apply against intentional changes carried out by special sophisticated software tools.</td></tr>
<tr><td colspan="3">Required Documentation:
The method of how the protection is realised and how corrupted data is marked shall be documented.</td></tr>
<tr><td colspan="2">Validation Guidance:
Checks based on documentation:
• If a checksum or signature is used:
Check that the checksum or signature is generated over the entire data set.
Check that legally relevant software, which reads the data and calculate a checksum or decrypts a signature really compares calculated and the nominal values.
• Check that secret data (e.g. key initial value if used) are kept secret against spying out with simple tools.</td><td>Validation Guidance (in addition to the guidance for risk classes B and C):
Checks based on documentation:
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="2">Example of an Acceptable Solution:
Stored data shall be accompanied by additional redundant information to enable the software retrieving, evaluating, and indicating or otherwise processing the data.

Just before the data is reused, the value of the checksum is recalculated and compared with the stored nominal value. If the values match, the data set is valid and may be used; otherwise it shall be deleted or marked invalid.
An acceptable solution is the CRC-16 algorithm.

<u>Note:</u> The algorithm is not secret but in contrast to requirement L2, the initial vector of the CRC-register or the generator polynomial (i.e. the divisor in the algorithm) must be. The initial vector and generator polynomial are known only to the programs generating and verifying the checksums. They shall be treated as keys (see L5).</td><td>Example of an Acceptable Solution:
Stored data shall be accompanied by additional redundant information to enable the software retrieving, evaluating, and indicating or otherwise processing the data.

Instead of the CRC, a signature is calculated. A suitable signature algorithm would be one of the hash algorithms, in combination with an encryption algorithm.

Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).

Note: Even if the algorithm and key meet the level high, a technical solution with a standard personal computer would not realise this protection level provided that there are no appropriate protection means for the programs that sign or verify a data set (see basic guide U for universal computers, comment on requirement U6-Risk Class D).</td></tr>
<tr><td colspan="3">Additions for Risk Class E</td></tr>
<tr><td colspan="3">Required Documentation (in addition to the documentation required for risk classes B to D):
Source code of the llegally relevant software that realises the integrity of stored data.</td></tr>
<tr><td colspan="3">Validation Guidance (in addition to the guidance for risk class D):
Checks based on the source code:
• Check whether measures taken for ensuring integrity are appropriate and correctly implemented.</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D
L4 Authenticity of measurement data stored *The measurement data stored shall be capable of being authentically traced back to the measurement that generated them.*		
Specifying Notes: 1. The authenticity of measurement data may be needed for reference at a later date, e.g., for checking invoices. 2. Authenticity requires the correct assignment (linking) of measurement data to the measurement that has generated the data. 3. Authenticity presupposes an identification of data sets. 4. Ensuring authenticity does not necessarily require an encryption of the data.		
Required Documentation: Description of the method used for ensuring the authenticity.		
Validation Guidance: *Checks based on documentation:* • Check that there is a correct linking between each measurement value and the corresponding measurement. • If a checksum or signature is used, check that the checksum or signature is generated over the entire data set. • Check that secret data (e.g. key initial value if used) are kept secret against spying out with simple tools. *Functional checks:* • Check whether corresponding stored data and data printed on the ticket or invoice are identical. • Check whether the ticket shows a hint that the measurement values can be compared with the reference data on a means of storage subject to legal control.		**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Example of an Acceptable Solution: A stored data set contains the following data fields (additional to the fields defined in L3): • A unique (sequential) identification number. The identification number is also copied to the delivery note. • Time when the measurement has been performed (time stamp). The time stamp is also copied to the delivery note. • An identification of the measuring instrument that has generated the value. • A signature that is used for ensuring the integrity of data can simultaneously be used for ensuring the authenticity. The signature covers all of the fields of the data set. Refer to requirement L2, L3. The ticket may state that the measurement values can be compared with the reference data on a means of storage subject to legal control. Assignment is demonstrated by comparing the identification number or time stamp printed on the delivery note with that in the stored data set.		**Example of an acceptable solution:** The origin of public keys used for signing the measurement data is verified by means of a PKI.

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code that generates the data sets for storing and realises the authentication..
Validation Guidance (in addition to the guidance for risk class D)**:** *Checks based on the source code:* • Check whether the data sets are correctly built and reliably authenticated.

Risk Class B	Risk Class C	Risk Class D
L5: Confidentiality of keys *Keys and associated information shall be treated as measurement data and shall be kept secret and be protected against compromise.*		

Specifying Notes:

1. This requirement only applies if a secret key is used at all.
2. This requirement only applies to measurement data storages, which are external from the measuring instrument or realised on universal computers.
3. If the access to the secret keys is prevented by hardware means, no additional software protection means are necessary.
4. The protection shall apply against intentional changes carried out by easily available and manageable software tools.

	5. The protection shall also apply against intentional changes carried out by special sophisticated software tools.

Required Documentation:

Description of the key management and means for keeping keys and associated information secret.

Validation Guidance: *Checks based on documentation:* • Check that the secret information cannot be compromised.	**Validation Guidance** (in addition to the guidance for risk classes B and C)**:** *Checks based on documentation:* • Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.
Example of an Acceptable Solution: The secret key and associated information are stored in binary format in the executable code of the legally relevant software. It is then not obvious at which address this information is stored. The system software does not offer any features to view or edit these data. If the CRC algorithm is used instead of a signature algorithm, the initial vector or generator polynomial plays the role of a key.	**Example of an Acceptable Solution:** The secret key is stored in a hardware part that can be physically sealed. The software does not offer any features to view or edit these data. Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)). *Note:* A technical solution with a standard personal computer would not be sufficient to ensure high protection level if there were no appropriate hardware protection means for the key and other secret data (see basic guide for universal computers U6). 1) *Public Key Infrastructure:* The public key of the storage subject to legal control has been certified by an accredited Trust Centre. 2) *Direct Trust:* It is not necessary to involve a trust centre if, by prior agreement, both parties, are able to read the public key of the measuring instrument directly at a device subject to legal control that is generating the relevant data set.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B to D)**:**
Source code that realises key management.

Validation Guidance (in addition to the guidance for risk class D)**:**

Checks based on the source code:

- Check whether measures taken for key management are appropriate.

Risk Class B	Risk Class C	Risk Class D
L6: Retrieval, verification, and indication of stored data *There shall be legally relevant software for displaying or printing measurement data stored.*		
Specifying Notes: 1. The software shall have the capability to display or print the measurement data stored along with the relevant information (see L1). 2. Retrieved data should be verified. 3. Displayed or printed measurement data shall indicate an eventual violation of authenticity and integrity.		
Required Documentation: • Description of the functions of the retrieval software. • Description how corrupted data is indicated.		
Validation Guidance: *Checks based on documentation:* • Check that the retrieval software has the required capabilities *Functional checks:* • Perform spot checks verifying that retrieval provides all necessary information.		
Example of an Acceptable Solution: 1. The data set is read from the storage by the retrieval software and the signature over all data fields is recalculated and compared with the stored nominal value. If both values match, the data set is correct, otherwise the data is deleted or marked as invalid by the program. 2. The measurement data stored might need to be referred to at a later date, e.g. transactions that are queried. If there is a doubt on the correctness of a delivery note or ticket, it must be possible to identify the measurement data stored to the disputed measurement without ambiguities (refer also to L1, L3, L4 and L5). 3. The identification number (see L1) must be printed out on the delivery note/ticket for the customer along with an explanation and a reference to the storage subject to legal control.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the retrieval software.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for retrieval, verification of signatures etc. are appropriate and correctly implemented.

Risk Class B	Risk Class C	Risk Class D
L7: Automatic storing *The measurement data shall be stored automatically when the measurement is concluded.*		
Specifying Notes: 1. The storing function shall not depend on the decision of the operator. 2. In cases where a decision is required from the operator whether or not to accept a measurement result, the measurement data shall be stored automatically after taking the decision.		
Required Documentation: Description of automatic storing. Description of the Graphical User Interface in case of operator-depended storing decisions.		
Validation Guidance: *Checks based on documentation:* • Check that storing process is automatic. *Functional checks:* • Examine by spot checks that the measurement values are stored automatically after measurement or acceptance of measurement is concluded. Check that there are no buttons or menu items to interrupt or disable the automatic storing.		
Example of an Acceptable Solution: There is no menu item or button in the Graphical User Interface (GUI) that supports manual initiation of storing measurement results. The measurement values are wrapped in a data set along with additional information such as time stamp and signature and are stored immediately after the measurement, or the acceptance of measurement, respectively.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for automatic storing are appropriate and correctly implemented.

Risk Class B	Risk Class C	Risk Class D

L8: Storage capacity and continuity

The long-term storage shall have a capacity which is sufficient for the intended purpose.

Specifying Notes:

1. When storage is full or removed or disconnected from the instrument, a warning shall be given to the operator. A warning is not necessary, if it is assured by construction that only outdated data can be overwritten.
2. The regulations concerning the minimum period for storing measurement data and the required inscriptions are left to national regulations and therefore beyond the scope of this guide.
3. The information on the capacity of the storage shall be made available.

Required Documentation:

Capacity of storage, Description of the management of storing measurement data.

Validation Guidance:

Checks based on documentation:

- Check that the capacity of storage or a formula for calculating it, is given.
- Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.

Functional checks:

- Check that a warning is given if the storage is full or removed, if applicable.

Example of an Acceptable Solution:

- For interruptible measurements that can be stopped easily and rapidly, e.g. weighing, fuel measurement, etc, the measurement may be completed even if the storage becomes unavailable. The measuring instrument or the device should have a buffer that is large enough to store the current transaction. After this, no new transaction may be started and the buffered values are kept for later transmission to a fresh storage.
- Measurements that are not interruptible, e.g. the measurement of energy, volume, etc, do not need a special intermediate buffer because these measurements always are cumulative. The cumulative register can be read out and transmitted to the storage at a later time when the storage is available again.
- Measurement data may be automatically overwritten by a tool that checks if the measurement data is out-of-date (refer to national regulations for the relevant time period) or that the invoice has been paid. The tool shall prompt the user for permission to delete and data shall be deleted in the order oldest first.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B to D)**:**

Source code of the legally relevant software that realises storing of data.

Validation Guidance (in addition to the guidance for risk classes B to D)**:**

Checks based on the source code:

- Check whether measures taken for storing are appropriate and correctly implemented.

7 Extension T: Transmission of Measurement Data via Communication Networks

The specific requirements of this section only apply if measurement data is transmitted via communication networks to a distant device where it is further processed and/or used for legally relevant purposes. They are an addition to the specific requirements of software for built-for-purpose measuring instrument (type P requirements) and of software for measuring instruments using a universal computer (type U requirements).

This extension does not apply if there is no subsequent measurement data processing. If software is downloaded to a device subject to legal control, then the requirements of Extension D apply.

7.1 Technical description

In the following table two network configurations are identified.

Description of configurations
A) Closed network Only a fixed number of participants with clear identity, functionality and location are connected. All devices in the network are subject to legal control.
B) Open network Arbitrary participants (devices with arbitrary functions) can be connected to the network. The identity and functionality of a participating device and its location may be unknown to other participants. Any network that contains legally controlled devices with infrared or wireless network communications interfaces shall be considered to be an open network.

Table 7-1: Technical description of communication networks.

7.2 Specific software Requirements for Data Transmission

8 Risk Class B	9 Risk Class C	10 Risk Class D
11 T1: Completeness of transmitted data *The transmitted data shall contain all relevant information necessary to present or further process the measurement result in the receiving unit.*		
Specifying Notes: 1. The completeness depends individually from the type of measurement.		
Required Documentation: Document all fields of the data set.		
Validation Guidance: *Checks based on documentation:* • Check whether all information for further processing the measurement values at the receiving unit are contained in the data set.		
Example of an Acceptable Solution: The data set comprises the following fields: • Measurement value(s) with correct resolution • the legally correct unit of measure • the unit price or the price to pay (if applicable) • the time and date of the measurement (if applicable) • identifier of the instrument if applicable (data transmission) • the place of the measurement (if applicable)		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code that generates the data sets for transmission.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether data sets are built correctly.

Risk Class B	Risk Class C	Risk Class D
T2: Protection against accidental or unintentional changes *Transmitted data shall be protected against accidental and unintentional changes.* **Specifying Notes:** 1. Means shall be implemented to protect from unintentional change or deletion of measurement data. **Required Documentation:** Description of the methods used to detect transmission errors. **Validation Guidance:** *Checks based on documentation:* • Check that a method is implemented to detect transmission errors. **Example of an Acceptable Solution:** • Transmitted data shall be accompanied by additional redundant information to enable the software of the receiver to detect accidental data transmission errors. • **To** detect data changes, a checksum with the **CRC-16** algorithm is calculated over all bytes of a data set and inserted into the data set to be transmitted. Just before the data is reused, the value of the checksum is recalculated by the receiver and compared with the attached nominal value. If the values match, the data set is valid and may be used, otherwise it shall be deleted or marked invalid. *Note:* The algorithm is not secret and, in contrast to requirement T3, neither is the initial vector of the CRC-register nor the generator polynomial i.e. the divisor in the algorithm. The initial vector and generator polynomial are known to both of the programs that create and verify the checksums. • Use of means provided by transmission protocols e.g. TCP/IP, IFSF.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software that realises the protection of transmitted data.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for protecting transmitted data are appropriate and correctly implemented.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">T3: Integrity of data
The transmitted measurement data shall be protected against intentional changes.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. This requirement only applies to open networks, not to closed networks.
2. The protection shall apply against intentional changes carried out by easily available and manageable software tools.</td></tr>
<tr><td colspan="2"></td><td>3. The protection shall also apply against intentional changes carried out by special sophisticated software tools</td></tr>
<tr><td colspan="3">Required Documentation:
Description of the protection method.</td></tr>
<tr><td colspan="3">Validation Guidance:
Checks based on documentation:
Check that an appropriate method has been selected.</td></tr>
<tr><td colspan="2">Example of an Acceptable Solution:
• Transmitted data shall be accompanied by additional redundant information to enable the software of the receiver to detect accidental data transmission errors.
• A checksum is generated of the data set to be transmitted. Just before the data is reused, the value of the checksum is recalculated and compared with the nominal value that is contained in the received data set. If the values match, the data set is valid and may be used, otherwise it shall be deleted or marked invalid.
• An acceptable solution is the CRC-16 algorithm.

<u>Note:</u> The algorithm is not secret but in contrast to requirement T2, the initial vector of the CRC-register or the generator polynomial (i.e. the divisor in the algorithm) is secret. The initial vector and generator polynomial are known only to the programs generating and verifying the checksums. They shall be treated as keys (see T5).</td><td>Example of an Acceptable Solution:
• Transmitted data shall be accompanied by additional redundant information to enable the software of the receiver to detect accidental data transmission errors.
• Instead of the CRC, a signature is calculated. Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).
• Protection is provided by some transmission protocols, e.g. HTTPS, TLS.

<u>Note:</u> To meet the high level of <u>protection</u>, appropriate protection means for the software (e.g., hardware support) that signs or verifies a data set are necessary (see also chapter 5 for software on universal computers, special requirement U6, specifying note 6 for risk class D).</td></tr>
<tr><td colspan="3">Additions for Risk Class E</td></tr>
<tr><td colspan="3">Required Documentation (in addition to the documentation required for risk classes B to D):
Source code of the legally relevant software that realises the integrity of transmitted data.</td></tr>
<tr><td colspan="3">Validation Guidance (in addition to the guidance for risk classes B to D):
Checks based on the source code:
• Check whether measures taken for guaranteeing integrity of transmitted data are appropriate.</td></tr>
</table>

Risk Class B	Risk Class C	Risk Class D
T4: Authenticity of transmitted data *The authenticity of transmitted measurement data shall be ensured.*		
Specifying Notes: 1. This requirement only applies to open networks, not to closed networks. 2. The protection shall apply against intentional changes carried out by easily available and manageable software tools. 3. The assignment of measurement values to a certain measurement shall be ensured.		
		~~4.~~ The protection shall also apply against intentional changes carried out by special sophisticated software tools.
Required Documentation: • Description of the authentication means.		
Validation Guidance: *Checks based on documentation:* Check that authentication means are adequate.		
Example of an Acceptable Solution: • Each data set has a unique (sequential) identification number, containing the date when the measurement has been performed (time stamp). • Each data set contains information about the origin of the measurement data, i.e. serial number or identity of the measuring instrument that generated the value. • In open networks, authenticity is guaranteed if the data set carries an unambiguous signature. The signature covers all of these fields of the data set. • The receiver of the data set checks all data for plausibility.		
		Example of an Acceptable Solution: Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of legally relevant software for sending and receiving device.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for guaranteeing the authenticity of transmitted data are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">T5: Confidentiality of keys
Keys and associated information shall be treated as measurement data and shall be kept secret and be protected against compromise.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. This requirement only applies if a secret key is used at all.
2. The protection shall apply against intentional changes carried out by easily available and manageable software tools.
3. If the access to the secret keys is prevented by hardware means, no additional software protection means are necessary.</td></tr>
<tr><td colspan="2"></td><td>4. The protection shall apply against intentional changes carried out by special sophisticated software tools.</td></tr>
<tr><td colspan="3">Required Documentation:
Description of the key management and means for keeping keys and associated information secret.</td></tr>
<tr><td colspan="2">Validation Guidance:
Checks based on documentation:
• Check that the secret information cannot be compromised.</td><td>Validation Guidance (in addition to the guidance for risk classes B and C):
Checks based on documentation:
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="2">Example of an Acceptable Solution:
The secret key and associated information are stored in binary format in the executable code of the legally relevant software. It is then not obvious at which address this information is stored. The system software does not offer any features to view or edit these data. If the CRC algorithm is used instead of a signature algorithm, the initial vector or generator polynomial play the role of a key.</td><td>Example of an Acceptable Solution:
The secret key is stored in a hardware part that can be physically sealed. The software does not offer any features to view or edit these data.

Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).

<u>Note:</u> A technical solution with a standard personal computer would not be sufficient to ensure high protection level if there were no appropriate hardware protection means for the key and other secret data (see basic guide for universal computers U6).
1) Public Key Infrastructure: The public key of the transmitting device subject to legal control has been certified by an accredited Trust Centre.
2) Direct Trust: It is not necessary to involve a trust centre if, by prior agreement, both parties, are able to read the public key of the measuring instrument directly at a device subject to legal control that is generating the relevant data set.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of legally relevant software that realises key management.
Validation Guidance (in addition to the guidance for risk class D)**:** *Checks based on the source code:* • Check whether measures taken for key management are appropriate.

Risk Class B	Risk Class C	Risk Class D
T6: Handling of corrupted data *Data that are detected as having been corrupted shall be marked to enable further processing software to react accordingly.*		
Specifying Notes: 1. Though communication protocols normally repeat a data transmission until it succeeds, nevertheless it is possible that a corrupted data set is received.		
Required Documentation: Description of the detection of corrupted data.		
Validation Guidance: *Checks based on documentation and functional checks:* • Check that corrupted data is detected and marked.		
Example of an Acceptable Solution: When the program that is receiving data sets detects a discrepancy between the data set and the nominal value of the signature, it first tries to reconstruct the original value if redundant information is available. If reconstruction fails, it generates a warning to the user, does not output the measurement value and • Sets a flag in a special field of the data set (status field) with the meaning “not valid”		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of legally relevant software in the receiving device.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for handling corrupted data are appropriate.

Risk Class B	Risk Class C	Risk Class D
T7: Transmission delay *The measurement shall not be inadmissibly influenced by a transmission delay.*		
Specifying Notes: 1. The timing of the data transmission shall be organised that under worst case conditions the measurement is not inadmissibly influenced.		
Required Documentation: Description of the concept, how measurement is protected against transmission delay.		
Validation Guidance: • Check the concept that the measurement is not influenced by transmission delay.		
Example of an Acceptable Solution: Implementation of transmission protocols for field buses.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B, C and D)**:** Source code of legally relevant software that realises the data transmission.
Validation Guidance (in addition to the guidance for risk classes B, C and D)**:** *Checks based on the source code:* • Check whether measures taken for handling transmission delay are appropriate.

Risk Class B	Risk Class C	Risk Class D
T8: availability of transmission services *If network services become unavailable, no measurement data shall get lost.*		
Specifying Notes: 1. It shall not be possible to corrupt measurement data by delaying or suppressing transmission. 2. The sending device shall be able to handle transmission disturbances accidentally happening 3. The reaction of the measuring instrument if transmission services become unavailable depends on the measuring principle (see Extension I).		
Required Documentation: Description of protection measures against transmission interruption or other failures.		
Validation Guidance: *Checks based on documentation:* • Check the measures taken to protect measurement data from transmission disturbances and interruption. *Functional checks:* • Spot checks shall show that no relevant data get lost due to a transmission interruption.		
Example of an Acceptable Solution: 1) For interruptible measurements that can be stopped easily and rapidly, e.g. weighing, fuel measurement, etc, the measurement may be completed even though the transmission is down. However, the measuring instrument or the device that is transmitting the measurement data has a buffer that is large enough to store the current transaction. After this no new transaction may be started and the buffered values are kept for later transmission. For other examples see part I. 2) Measurements that are not interruptible, e.g. the measurement of energy, volume, etc, do not need a special intermediate buffer because these measurements always are cumulative. The cumulative register can be read out and transmitted at a later time when the connection is up again.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software that realises data transmission.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for reacting on interrupted transmission service are appropriate.

8 Extension S: Software Separation

Software separation is an optional design method that allows to separate legally relevant software from legally non-relevant software. The communication between these parts of software is carried out via controlled interfaces. If following the conditions for software separation, the manufacturer need not to pass conformity assessment procedures when changing legally non-relevant software.

The specific requirements of this extension, if applicable, shall be considered in addition to the basic requirements of types P or type U instruments, respectively, described in Chapters 4 and 5 of this guide.

8.1 Technical description

Software controlled measuring instruments or systems in general have complex functionality and contain modules that are legally relevant and modules that are not. It is advantageous – though it is not prescribed – to separate these types of software modules.

In the following table, two variants of software separation are described. Both variants are covered by the subsequent set of special requirements.

Description
Software separation is realised independently from the operating system within an application domain, i.e., at the *programming language level (**Low level software separation**)*. *Note: **Low level separation:** Merging software units on the level of the programming language or merging parts of a programme (i.e. subroutines, procedures, functions, classes) to form the legally relevant part of the programme. The rest of the programme is the legally non-relevant part.* This feature is realisable in both built-for-purpose devices and universal computers.
The software modules to be separated are realised as independent objects in terms of the *operating system (**High level software separation**)*. *Note: **High level separation:** Merge all parts of the software to one object that is identifiable by the operating system (a programme, a DLL etc). The rest of the software is the legally non-relevant part.*) This type of separation is typical for software on universal computers. Example solutions are independently executable programs, dynamically linked libraries etc.

Table 8-1: Technical description of software separation

8.2 Specific software requirements for software separation

Risk Class B	Risk Class C	Risk Class D
S1: Realisation of software separation *There shall be a part of the software that contains all legally relevant software and parameters that is clearly separated from other parts of software.*		
Specifying Notes: 1. In the case of *low level separation,* all *program units* (subroutines, procedures, functions, classes, etc.) and in case of *high level separation* all *programs and libraries* ° that contribute to the calculation of measurement values or have an impact on it, ° that contribute to auxiliary functions such as displaying data, data security, data storage, software identification, performing software download, data transmission or storing, verifying received or stored data etc. belong to the legally relevant software. All *variables, temporary files and parameters* that have an impact on measurement data or on legally relevant software also belong to the legally relevant software. 2. The protective software interface itself (see S3) is part of the legally relevant software. 3. Legally non-relevant software comprises the remaining program units, data or parameters not covered above.		
Required Documentation: Naming of all components that belong to the legally relevant software.		
Validation Guidance: *Checks based on documentation:* • Check that the naming is correct and the list of named components is complete.		
Example of an Acceptable Solution:		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check (e.g. by data flow analysis with tools or manually) that all program units, programs or libraries that are involved in processing the measurement values are registered as legally relevant software.

Risk Class B	Risk Class C	Risk Class D
S2: Mixed indication *Information generated by the legally non-relevant software shall be shown on a display or printout in a way that confusions with the information generated by the legally relevant software are avoided.*		
Specifying Notes: ---		
Required Documentation: Description of the legally relevant software that realises the indication. Description of how the indication of legally relevant information is protected against misleading indication generated by legally non-relevant software.		
Validation Guidance: *Functional checks:* • Judge through visual check that additional information generated by legally non-relevant software and presented on display or printout cannot be confused with the information originating from legally relevant software.		
Example of an Acceptable Solution: • If additional information, part which is legally not relevant, should be indicated besides the legally relevant e.g. product identifier, an indication pattern shall be defined which is controlled by the legally relevant software. To ensure that all legally relevant information is extracted from an input string, it should pass through a filter which is part of the legally relevant software that detects inadmissible information, e.g. measurement units. The admissible information is then inserted into the indication pattern controlled by the legally relevant software.		

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check that legally relevant software generates the indication of measurement values. • Check whether the realised implementation of mixed indication is correct. • Check that this indication cannot be changed or suppressed by legally non-relevant programs.

Risk Class B	Risk Class C	Risk Class D

S3: Protective software interface

The data exchange between the legally relevant and legally non-relevant software shall be exclusively carried out via a protective software interface.

Specifying Notes:

1. This requirement applies to all kind of interactions and data exchanges between the legally relevant and legally non-relevant software.
2. All communication shall exclusively be carried out via the defined protective interface.
3. There shall be only those interactions and data flows allowed that do not inadmissibly influence the measuring process, in particular the legally relevant software, device-specific parameters and measurement data.
4. Scheduling and runtime of the measuring process shall not be influenced by legally non-relevant software

Required Documentation: Description of the software interface

- Description of the interface including description of allowed interactions and data flows.

Validation Guidance:

Checks based on documentation:

- Check that functions of the legally relevant software and actions of the measuring process, that may be triggered via the protective software interface are defined and described.
- Check that data that may be exchanged via the interface are defined and described.
- Undertake plausibility checks that the description of interactions and data exchanges is complete.

Example of an Acceptable Solution:

- The data domains of the legally relevant software part are encapsulated by declaring only local variables in the legally relevant part.
- The interface is realised as a subroutine belonging to the legally relevant software that is called from the legally non-relevant software. The data to be transferred to the legally relevant software are passed as parameters of the subroutine.
- The legally relevant software filters out inadmissible interface commands.

Additions for Risk Class E

Required Documentation (in addition to the documentation required for risk classes B to D):
Source code of the legally relevant software.

Validation Guidance (in addition to the guidance for risk classes B to D)**:**

Checks based on the source code:

- Check the software design whether data flow is unambiguously defined in the legally relevant software and can be verified.
- Check the data flow via the software interface by using appropriate tools or manually. Check whether the complete data flow between the software parts has been documented. Search for inadmissible data flow.
- Check that interactions triggered by the legally non-relevant software are documented. Search for inadmissible interactions.

9 Extension D: Download of Legally Relevant Software

This extension shall be used if instruments are equipped with facilities for a software download without breaking a seal. The specific requirements of this extension, if applicable, are to be considered in addition to the basic requirements of types P or type U instruments, respectively, described in Chapters 4 and 5 of this guide.

This guide does not impose any prescriptions whether a software download to instruments in use without breaking a seal is allowed or not. However, if a download without breaking a seal is allowed, then the specific requirements laid down below shall be considered.

9.1 Technical Description

The scope of configurations, which are in principle suitable for a software download is large. It is described in the following table.

Hardware Configuration The instrument with facilities for a software download may be a built-for-purpose type (Type P) or an instrument with a universal computer (Type U). Communications links for the software transmission may be direct, e.g. RS 232, USB, over closed networks, e.g. Ethernet, token-ring LAN, or over open networks, e.g. Internet.
Software Configuration The entire software to be downloaded may be legally relevant or there may be a separation between legally relevant and legally non-relevant software. In the latter case, only the download of legally relevant software is subject to the requirements laid down below. Download of legally non relevant software is allowed without any restrictions, provided the software separation has been certified.

Table 9-1: Technical description of configurations for automatic software download.

The software download consists of two (logical) phases: (1) The transmission process to the measuring instrument and (2) the installation of the software transmitted.

9.2 Specific Software Requirements

Risk Class B	Risk Class C	Risk Class D

D1: Download mechanism

Both phases of the software download, the transmission and the subsequent installation of software, shall run automatically and not affect the protection of legally relevant software.

Specifying Notes:

1. The instrument shall be equipped with legally relevant software that carries out the checking functions required in D2 to D4.
2. The instrument shall be capable of detecting if the transmission of software or the subsequent installation fails. A warning shall be given. If the transmission or the installation is unsuccessful or has been interrupted, then the original status of the measuring instrument shall be unaffected. Alternatively, the instrument shall display a permanent error message and its metrological functioning shall be inhibited until the fault has been cleared.
3. On successful completion of the installation, all protective means shall be activated.
4. During transmission and subsequent installation of software, the measurement process shall be inhibited or correct measurement shall be appropriately guaranteed.
5. The number of retries of transmissions and installation attempts shall be reasonably limited.

Required Documentation:

The documentation shall describe how the conditions given in the specifying notes are implemented.

Validation Guidance:

Check that the conditions given in the specifying notes are fulfilled.

Functional checks:

- Perform at least one software download to check its correct process.

Example of an Acceptable Solution:

The whole legally relevant software part is fixed, i.e. it cannot be downloaded or changed without breaking a seal.

An auxiliary program resident in the legally relevant part of the software that:

a. Handshakes with the sender and checks for consent
b. Automatically inhibits measurement during transmission and installation
c. Automatically transmits the legally relevant software to a secure holding area
d. Automatically carries out the checks required by D2 to D4
e. Automatically installs the software into the correct location
f. Takes care of housekeeping, e.g. deletes redundant files, etc.
g. Ensures that any protection removed to facilitate transmission and installation is automatically replaced to the required level on completion.
h. Initiates the appropriate fault handling procedures if a fault occurs.

For member states where software download for instruments in use is not allowed, it shall be possible to disable the software download mechanism by means of a sealable setting (switch, secured parameter). In this case it must not be possible to download legally relevant software without breaking the seal.

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Part of source code of legally relevant software that is responsible for the management of the download process.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for managing the download process are appropriate.

Risk Class B	Risk Class C	Risk Class D
D2: Authentication of transmitted software *Means shall be employed to guarantee that the transmitted software is authentic.* **Specifying Notes:** 1. Before the transmitted software is installed, it shall be checked that: a. The software is authentic. b. The software belongs to the measuring instrument on which it shall be installed. 2. A negative check result shall be considered as failure of transmission and treated as laid down in D1.		
Required Documentation: The documentation shall describe how the checks mentioned in the specifying notes are carried out.		
Validation Guidance: *Checks based on documentation:* • Check that the described checks are appropriate *Functional checks:* • Check that installation of not authentic or not to the instrument belonging software is inhibited.		
Example of an Acceptable Solution: 1. **Authenticity:** For integrity reasons (see D3) an electronic signature is generated over the software part to be downloaded. Authenticity is guaranteed if a key stored in the legally relevant software of the instrument confirms that the signature originates from the authorised body. Signature matching is done automatically. The key can only be exchanged by breaking a seal. 2. **Correct type of measuring instrument** Checking the instrument type requires automatically matching an identification of instrument type that is stored in the legally relevant software part of the instrument with a compatibility list attached to the software.		
		Example of an Acceptable Solution: Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of the legally relevant software part that is responsible for checking the authenticity.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures are taken for checking the conditions laid down in the specifying notes.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">D3: Integrity of downloaded software
Means shall be employed to guarantee that the software has not been changed during transmission.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. Before the transmitted software is installed, it shall be checked that the software has not been changed during transmission.
2. A negative check result shall be considered as failure of transmission and treated as laid down in D1.</td></tr>
<tr><td colspan="3">Required Documentation:
The documentation shall describe how the checks are carried out.</td></tr>
<tr><td colspan="3">Validation Guidance:
Checks based on documentation:
• Check that the described check is appropriate.
Functional checks:
• Check that installation of changed software is inhibited.</td></tr>
<tr><td colspan="2">Example of an Acceptable Solution:
• Integrity is demonstrated by calculating a checksum over the legally relevant software and comparing it against the checksum attached to the software.
• Acceptable algorithm: CRC, secret initial vector, length 32 bit. The initial vector is stored in the legally relevant software part.</td><td>Example of an Acceptable Solution:
• A suitable signature algorithm (e.g. SHA with RSA). Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).
• The key for decrypting is stored in the legally relevant software part and cannot be exchanged or read out without breaking a seal.</td></tr>
</table>

Additions for Risk Class E
Required Documentation (in addition to the documentation required for risk classes B to D)**:** Source code of that part of legally relevant software that is responsible for checking the integrity of the software.
Validation Guidance (in addition to the guidance for risk classes B to D)**:** *Checks based on the source code:* • Check whether measures taken for checking the integrity are appropriate.

<table>
<tr><th>Risk Class B</th><th>Risk Class C</th><th>Risk Class D</th></tr>
<tr><td colspan="3">D4: Traceability of legally relevant software download
It shall be guaranteed by appropriate technical means that downloads of legally relevant software are adequately traceable within the instrument for subsequent controls.</td></tr>
<tr><td colspan="3">Specifying Notes:
1. All relevant data making a download or a download attempt traceable shall be recorded and secured. Relevant data includes date and time of download, identifier(s) of software, origin of transmission, success note.
2. The data recorded shall be available for an adequate period of time (the period depends on regulations outside MID).
3. The recorded data shall be presented on demand.
4. The traceability means and records are part of the legally relevant software and shall be protected as such.</td></tr>
<tr><td colspan="3">Required Documentation:
The documentation shall describe:
• how the traceability means are implemented and protected,
• the structure of records,
• how the recorded data may be presented.</td></tr>
<tr><td colspan="2">Validation Guidance:
Checks based on documentation:
• Check that implemented traceability means fulfil the conditions laid down in the specifying notes.
Functional checks:
• Check the functionality of the means while carrying out a software download.</td><td>Validation Guidance (in addition to the guidance for risk classes B and C):
Checks based on documentation:
• Check whether the measures taken are appropriate with respect to the required state of the art for a high protection level.</td></tr>
<tr><td colspan="3">Example of an Acceptable Solution:
• Event logger. The measuring instrument is equipped with an event logger that automatically records at least the date and time of the download, identifier of the downloaded legally relevant software, the identifier of the downloading party, and an entry of the success. An entry is generated for each download attempt regardless of the success.
• After having reached the limit of the event logger, it is ensured by technical means that further downloads are impossible. Event logger may only be erased by breaking a seal and may be resealed only by the inspection authorities.</td></tr>
<tr><td colspan="2"></td><td>Example of an Acceptable Solution:
Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)).</td></tr>
</table>

<table>
<tr><th>Additions for Risk Class E</th></tr>
<tr><td>Required Documentation (in addition to the documentation required for risk classes B to D):
Source code of the legally relevant software part that is responsible for tracing download processes.</td></tr>
<tr><td>Validation Guidance (in addition to the guidance for risk class D):
Checks based on the source code:
• Check whether measures taken for tracing the download process are appropriate.
• Check whether measures taken for protecting the recorded data are appropriate.</td></tr>
</table>

10 Extension I: Instrument Specific Software Requirements

This extension is intended to complement the general software requirements of the previous chapters and cannot be considered isolated from parts P or U and the other extensions (see Chapter 2). It reflects the existence of instrument-specific MID annexes MI-x and contains specific aspects and requirements for measuring instruments or systems (or sub-assemblies). These requirements do not, however, go beyond the requirements of the MID. If reference is made to OIML recommendations or ISO/IEC standards this is done only if these can be considered as normative documents in the sense of the MID and if this supports a harmonised interpretation of the MID requirements.

Besides instrument specific software aspects and requirements Extension I contains the instrument (or category) specific assignment of risk classes which ensures a harmonised level of software examination, software protection and software conformity.

For the present, Extension I is intended to be an initial draft to be completed by the respective WELMEC Working Group that has the corresponding specific knowledge. Therefore Extension I has an "open structure", i.e. it provides a skeleton that is - besides the initial assignment of risk classes - filled-in only partly (e.g. for utility meters and automatic weighing instruments). It may be used for other MID (or non-MID) instruments, too, according to the experiences gained and decisions taken by the responsible WELMEC Working Groups. The numbering x of the sub-chapters 10.x follows the numbering of the specific MID Annex MI-x. Non-MID instruments could be added starting from 10.11.

There are different instrument specific software aspects that might need consideration for a certain type x of measuring instrument. These aspects should be treated in a systematic manner as follows: Each sub-chapter 10.x should be subdivided into sections 10.x.y where y covers the following aspects.

10.x.1 Specific regulations, standards and other normative documents

Here, instrument (or category) specific regulations, standards and other normative documents (e.g. OIML recommendations) or WELMEC guidelines should be mentioned that may help to develop instrument (or category) specific software requirements as an interpretation of the requirements of the MID Annex I and the specific annexes MI-x.

Normally the specific software requirements apply in addition to the general ones in the previous chapters. Otherwise it should be clearly stated whether a specific software requirement replaces one (or more) of the general software requirements, or whether one (or more) general software requirements is (are) not applicable, and the reason why.

10.x.2 Technical description

Here

- examples of most common specific technical configurations,
- the application of parts P, U and extensions to these examples, and
- useful (instrument specific) checklists for both the manufacturer and the examiner

may be given. The description should mention

- the measuring principle (cumulative measurement or single independent measurement; repeatable or non-repeatable measurement; static or dynamic measurement), and
- the fault detection and reaction; two cases are possible:
  a) the presence of a defect is obvious or can simply be checked or there are hardware means for fault detection,
  b) the presence of a defect is not obvious and cannot be easily checked and there are no hardware means for fault detection.

  In the latter case (b) fault detection and reaction requires appropriate software means and hence appropriate software requirements.
- the hardware configuration; at least the following issues should be addressed:
  a) Is there a modular, general-purpose computer-based system or a dedicated instrument with an embedded system subject to legal control?
  b) Does the computer system stand-alone, or is it part of a closed network, e.g. Ethernet, token-ring LAN, or part of an open network, e.g. Internet?
  c) Is the sensor separated (separate housing and separate power supply) from the Type U system or is it partly or completely integrated into it?
  d) Is the user interface always under legal control (both for Type P and Type U instruments) or can it be switched to an operating mode which is not under legal control?
  e) Is long-term data storage foreseen? If yes, then is the storage local (e.g. hard disk) or remote (e.g. file server)?
  f) Is the storage medium fixed (e.g. internal ROM) or removable (e.g. floppy disc, CD-RW, smart-media card, memory stick)?
- the software configuration and environment; at least the following issues should be addressed:
  a) Which operating system is used or can be used?
  b) Do other software applications reside on the system besides the legally relevant software?
  c) Is there software not subject to legal control that is intended to be freely modified after approval?

10.x.3 Specific software requirements

Here, the specific software requirements should be listed and commented using a similar form as in the previous chapters.

10.x.4 Examples of legally relevant parameters, functions, and data

Here, examples of

- device specific parameters (e.g. individual configuration and calibration parameters of a specific measuring instrument),
- type specific parameters (e.g. specific parameters that are fixed at type examination), or
- legally relevant, specific functions

may be given.

10.x.5 Other aspects

Here, other aspects, e.g. specific documentation required for type (software) examination, specific descriptions, and instructions to be supplied in type examination certificates, or other aspects (e.g. requirements concerning the testability) may be mentioned.

10.x.6 Assignment of risk class

Here, the appropriate risk class for instruments of type x should be defined. This can be done

- either generally (for all categories within the respective type), or
- depending on the field of application, or category, or other aspects if these exist.

10.1 Water Meters

10.1.1 Specific regulations, standards and other normative documents

Member states may – in accordance with MID Article 2 – prescribe Water meters in residential, commercial and light industrial use to be subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-001 only.

OIML recommendations and standards have not been taken into consideration.

10.1.2 Technical description

10.1.2.1 Hardware Configuration

Water meters are typically realised as built-for purpose devices (Type P in this document).

10.1.2.2 Software Configuration

This is specific to each manufacturer but would normally be expected to follow the recommendations given in the main body of this guide.

10.1.2.3 Measuring Principle

Water meters continually cumulate the volume consumed. The cumulative volume is displayed at the instrument. Various principles are employed.

The volume measurement may not be repeated.

10.1.2.4 Fault Detection and Reaction

The requirement MI-001, 7.1.2 deals with electromagnetic disturbances. There is a need to interpret this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes no difference what the reason for a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.1.3 Specific software requirements (Water meters)

Risk Class B	Risk Class C	Risk Class D
I1-1: Fault Recovery *The software shall recover from a disturbance to normal processing.*		
Specifying Notes: Date stamped flags should be raised to help log periods of faulty operation.		
Required Documentation: A brief description of the fault recovery mechanism and when it is invoked.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault recovery is appropriate. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. If any function has not been processed or – in the worst case – the microprocessor hangs in an arbitrary endless loop, the reset of the watchdog does not happen and it fires after a certain time span.		

Risk Class B	Risk Class C	Risk Class D
I1-2: Back-up Facilities *There shall be a facility that provides for periodic back-up of measurement data, such as measurement values and the current status of the process. This data shall be stored in a non-volatile storage.*		
Specifying Notes: The storage intervals shall be sufficiently small so that the discrepancy between the current and saved cumulative values is small.		
Required Documentation: A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values.		
Validation Guidance: *Checks based on documentation:* • Check whether measurement data is saved to non-volatile storage and can be recovered. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: Measurement data is backed up as required (e.g. every 60 minutes)		

Risk Class B	Risk Class C	Risk Class D
I1-3: MID-Annex I, 8.5 (Inhibit resetting of cumulative measurement values) *For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.*		
Specifying Notes: Cumulative registers of a measuring instrument may be reset prior to being put into use.		
Required Documentation: Documentation of protection means against resetting the volume registers.		
Validation Guidance: *Checks based on documentation:* • Check that cumulative legally relevant measurement values cannot be reset without leaving a trace. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: The registers for volume are protected against changes and resetting by the same means as parameters (see P7).		

Risk Class B	Risk Class C	Risk Class D
I1-4: Dynamic behaviour *The legally non-relevant software shall not adversely influence the dynamic behaviour of a measuring process.*		
Specifying notes: • This requirement applies in addition to S-1, S-2 and S-3 if software separation has been realised in accordance with extension S. • The additional requirement ensures that for real time applications of meters the dynamic behaviour of the legally relevant software is not inadmissibly influenced by legally non-relevant software, i.e. the resources of the legally relevant software are not inadmissibly reduced by the legally non relevant part.		
Required Documentation: • Description of the interrupt hierarchy. • Timing diagram of the software tasks. Limits of proportionate runtime for legally non-relevant tasks.		
Validation Guidance: *Checks based on documentation:* • Documentation of the limits of the proportionate runtime for legally non-relevant tasks is available for the programmer of the legally non-relevant software part. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an acceptable solution: The interrupt hierarchy is designed in a way that avoids adverse influences.		

Risk Class B	Risk Class C	Risk Class D
I1-5: Imprinted Software Identifier *The software identifier is usually presented on a display. As an exception for water meters, an imprint of the software identifier on the name plate of an instrument shall be an acceptable solution if the following conditions A, B and C are fulfilled:* *A. The user interface does not have any control capability to activate the indication of the software identifier on the display or the display does not allow technically showing the identifier of the software (mechanical counter).* *B. The instrument does not have any interface to communicate the software identifier.* *C. After production of a meter a change of the software is not possible or only possible if also the hardware or a hardware part is changed.*		
Specifying notes: • The manufacturer of the hardware or the concerned hardware part is responsible that the software identifier is correctly marked on the concerned hardware. • All other Specifying Notes of P2/U2 apply.		
Required Documentation: • According to P2/U2.		
Validation Guidance: *Checks based on documentation:* • According to P2/U2. *Functional checks:* • According to P2/U2.		
Example of an acceptable solution: Imprint of the software identifier on the name plate of the instrument.		

10.1.4 Examples of legally relevant parameters, functions, and data

Water meters have parameters like constants for calculations, for configuration etc, but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirements P2 and P7, guide P.

In the following some typical parameters of water meters are given. (This table will be updated when WELMEC Working Group 11 has decided on the final contents.)

Parameter	Protected	Settable	Comment
Calibration factor	x		
Linearisation factor	x		

10.1.5 Other aspects

For domestic applications it is expected that download of software (Extension D, Chapter 9) will not be very important.

The cumulating energy or volume register of domestic instruments is not a long-term storage in the sense of Extension L (Chapter 6). For an instrument that only measures cumulated energy / volume the application of the extension L is not necessary.

10.1.6 Assignment of risk class

For the present, according to the decisions of the responsible WELMEC Working Group 11, the following risk class is considered appropriate and should be applied, if software examinations based on this guide are carried out for (software-controlled) water meters:

- **Risk class C for instruments of type P**

A final decision has, however, not yet been taken and WG 11 will reconsider this item in connection with the discussion of appropriate risk class(es) for type U instruments.

10.2 Gas Meters and Volume Conversion Devices

10.2.1 Specific regulations, standards and other normative documents

Member states may – in accordance with MID Article 2 – prescribe Gas meters and volume conversion devices in residential, commercial and light industrial use to be subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-002 only.

OIML recommendations and standards have not been taken into consideration.

10.2.2 Technical description

10.2.2.1 Hardware Configuration

Gas meters and volume conversion devices are typically realised as built-for purpose devices (Type P in this document). They may have one or more inputs for external sensor units and meter and conversion devices may be different hardware units.

10.2.2.2 Software Configuration

This is specific to each manufacturer but would normally be expected to follow the recommendations given in the main body of this guide.

10.2.2.3 Measuring Principle

Gas meters continually cumulate the volume consumed. The cumulative volume is displayed at the instrument. Various principles are employed. A volume converter is used to calculate the volume at base conditions. The converter may be an integral part of the meter.

The volume measurement may not be repeated.

10.2.2.4 Fault Detection and Reaction

The requirement MI-002, 4.3.1 deals with electromagnetic disturbances. There is a need to interpret this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes no difference what the reason for a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.2.3 Specific software requirements (Gas meters and volume converters)

Risk Class B	Risk Class C	Risk Class D
I2-1: Fault Recovery *The software shall recover from a disturbance to normal processing.*		
Specifying Notes: Date stamped flags should be raised to help log periods of faulty operation.		
Required Documentation: A brief description of the fault recovery mechanism and when it is invoked.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault recovery is appropriate. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog		

Risk Class B	Risk Class C	Risk Class D
I2-2: Back-up Facilities *There shall be a facility that provides for periodic back-up of measurement data, such as measurement values and the current status of the process. This data shall be stored in a non-volatile storage.*		
Specifying Notes: The storage intervals shall be sufficiently small so that the discrepancy between the current and saved cumulative values is small.		
Required Documentation: A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values.		
Validation Guidance: *Checks based on documentation:* • Check whether measurement data is saved to non-volatile storage and can be recovered. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: Measurement data is backed up as required (e.g. every 60 minutes)		

Risk Class B	Risk Class C	Risk Class D
I2-3: MI-002, 5.32 (indication suitability) *The display of the total uncorrected volume shall have a sufficient number of digits to ensure that when the meter is operated for 8000 hours at Q_{max}, the indication does not return to its initial value.*		
Specifying Notes: ---		
Required Documentation: Documentation of the internal representation of the volume register.		
Validation Guidance: *Checks based on documentation:* • Check whether storage capacity is sufficient.		
Example of an Acceptable Solution: Typical values for domestic gas meter are: Q_{max} = 6 m^3/h. The required range is 48000 m^3. (currently electronic gas meters display up to 99999m^3)		

Risk Class B	Risk Class C	Risk Class D
I2-4: MID-Annex I, 8.5 (Inhibit resetting of cumulative measurement values) *For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.*		
Specifying Notes: Cumulative registers of a measuring instrument may be reset prior to being put into use.		
Required Documentation: Documentation of protection means against resetting the volume registers.		
Validation Guidance: *Checks based on documentation:* • Check that cumulative legally relevant measurement values cannot be reset without leaving a trace. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: The registers for volume are protected against changes and resetting by the same means as parameters (see P7).		

Risk Class B	Risk Class C	Risk Class D
I2-5: MI-002, 5.2 (Power source lifetime) *A dedicated power source shall have a lifetime of at least five years. After 90% of its lifetime an appropriate warning shall be shown.*		
Specifying Notes: Lifetime is used here in the sense of available energy capacity. If the power source can be changed in the field, parameters and measurement data shall not be corrupted during the changeover.		
Required Documentation: Documentation of the power source capacity, maximum lifetime (independent of energy consumption), measures to determine the consumed or available energy, description of the means for the warning of low available energy.		
Validation Guidance: *Checks based on documentation:* • Check whether the measures taken are appropriate for the surveillance of the energy available.		
Example of an Acceptable Solution: The operating hours or the wake-up events of the device are counted, stored in a non-volatile memory and compared with the nominal value of the battery lifetime. If 90% of the lifetime has elapsed an appropriate warning is shown. The software detects the exchange of the power source and resets the counter. Another solution would be to monitor the health of the power supply continuously.		

Risk Class B	Risk Class C	Risk Class D
I2-6: MI-002, 9.1 (Electronic conversion device) *An electronic conversion device shall be capable of detecting when it is operating outside the operating range(s) stated by the manufacturer, for parameters that are relevant for measurement accuracy. In such a case, the conversion device shall stop integrating the converted quantity, and may totalise separately the converted quantity for the time it is operating outside the operating range(s).*		
Specifying Notes: There shall be a display indication of the failure state.		
Required Documentation: Documentation of the different registers for converted quantity and failure quantity.		
Validation Guidance: *Checks based on documentation:* • Check whether the measures taken are appropriate for the management of unusual operating conditions.		
Example of an Acceptable Solution: The software monitors the relevant input values and compares them with predefined limits. If all values are inside the limits the converted quantity is integrated to the normal register (a dedicated variable). Else it totalises the quantity in another variable. Another solution would be to have only one cumulating register but to record the start and end date, time and register values of the out-of-range period in an event logger (see P7). Both quantities can be indicated. The user can clearly identify and distinguish the regular and the failure indication by means of a status indication.		

Risk Class B	Risk Class C	Risk Class D
I2-7: MI-002, 5.5 (Test element) *The gas meter shall have a test element, which shall enable tests to be carried out in a reasonable time.*		
Specifying Notes: The test element for accelerating time consuming test procedures is normally used for testing before installation and normal operation. During the test mode the same registers and software parts shall be used as during standard operating mode.		
Required Documentation: Documentation of the test element and instructions for activating the test mode.		
Validation Guidance: *Checks based on documentation:* • Check whether all time consuming test procedures of the gas meter can be completed by means of the test element.		
Example of an Acceptable Solution: The time base of the internal clock can be accelerated. Processes that last e.g. a week, a month or even a year and overrun of registers may be tested in the test mode within a time span of minutes or hours.		

Risk Class B	Risk Class C	Risk Class D
I2-8: Dynamic behaviour *The legally non-relevant software shall not adversely influence the dynamic behaviour of a measuring process.*		
Specifying notes: • This requirement applies in addition to S-1, S-2 and S-3 if software separation has been realised in accordance with extension S. • The additional requirement ensures that for real time applications of meters the dynamic behaviour of the legally relevant software is not inadmissibly influenced by legally non-relevant software, i.e. the resources of the legally relevant software are not inadmissibly reduced by the non-legal part.		
Required Documentation: • Description of the interrupt hierarchy. • Timing diagram of the software tasks. Limits of proportionate runtime for legally non-relevant tasks.		
Validation Guidance: *Checks based on documentation:* • Documentation of the limits of the proportionate runtime for legally non-relevant tasks is available for the programmer of the legally non-relevant software part. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an acceptable solution: The interrupt hierarchy is designed in a way that avoids adverse influences.		

Risk Class B	Risk Class C	Risk Class D
I2-9: Imprinted Software Identifier *The software identifier is usually presented on a display. As an exception for gas meters and volume converters, an imprint of the software identifier on the name plate of an instrument shall be an acceptable solution if the following conditions A, B and C are fulfilled:* *A. The user interface does not have any control capability to activate the indication of the software identifier on the display or the display does not allow technically showing the identifier of the software (mechanical counter).* *B. The instrument does not have any interface to communicate the software identifier.* *C. After production of a meter a change of the software is not possible or only possible if also the hardware or a hardware part is changed.*		
Specifying notes: • The manufacturer of the hardware or the concerned hardware part is responsible that the software identifier is correctly marked on the concerned hardware. • All other Specifying Notes of P2/U2 apply.		
Required Documentation: • According to P2/U2.		
Validation Guidance: *Checks based on documentation:* • According to P2/U2. *Functional checks:* • According to P2/U2.		
Example of an acceptable solution: Imprint of the software identifier on the name plate of the instrument.		

10.2.4 Examples of legally relevant parameters, functions, and data

Gas meters and volume converters often have a lot of parameters. They are used as constants for calculations, as configuration parameters etc, but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirements P2 and P7, guide P.

In the following some typical parameters of gas meters and volume conversion devices are given. (This table will be updated when WELMEC Working Group 11 has decided on the final contents.)

Parameter	Protected	Settable	Comment
Calibration factor	x		
Linearisation factor	x		

10.2.5 Other aspects

For domestic applications it is expected that download of software (extension D, Chapter 9) will not be very important.

The cumulating energy or volume register of domestic instruments is not a long-term storage in the sense of extension L (Chapter 6). For an instrument that only measures cumulated energy / volume the application of the extension L is not necessary.

10.2.6 Assignment of risk class

For the present, according to the decisions of the responsible WELMEC Working Group 11, the following risk class is considered appropriate and should be applied, if software

examinations based on this guide are carried out for (software-controlled) gas meters and volume conversion devices:

- **Risk class C for instruments of type P**

A final decision has, however, not yet been taken and WG 11 will reconsider this item in connection with the discussion of appropriate risk class(es) for type U instruments.

WG 11 considers prepayment and interval metering functionality to be additional to those essential measurement functions specified by MID Annex MI-002, therefore no greater risk category is allocated to these variants than to the basic meter types already covered by this software guide. However, the basic measurement function should be assessed, as with all other type P instruments along with any other assessment deemed necessary to demonstrate that the associated software providing these functions has no inadmissible influence on the basic measurement.

10.3 Active Electrical Energy Meters

10.3.1 Specific regulations, standards and other normative documents

Member states may – in accordance with MID Article 2 – prescribe Active electrical energy meters in residential, commercial and light industrial use to be subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-003 only.

OIML recommendations or IEC standards have not been taken into consideration.

10.3.2 Technical description

Active electrical energy meters take voltages and currents measurements as inputs, derive the active electrical power from them, and integrate this with respect to time to give the active electrical energy.

10.3.2.1 Hardware Configuration

Active electrical energy meters typically are realised as built-for purpose devices (Type P in this document). They may have one or more inputs and may be used in combination with external instrument transformers.

10.3.2.2 Software Configuration

This is specific to each manufacturer but would normally be expected to follow the recommendations given in the main body of this guide.

10.3.2.3 Measuring Principle

Active electrical energy meters continually cumulate the energy consumed in a circuit. The cumulative energy value is displayed at the instrument. Various transducer and multiplier principles are employed.

The energy measurement may not be repeated.

10.3.2.4 Fault Detection and Reaction

The requirement MI-003, 4.3.1 deals with electromagnetic disturbances. There is a need to interpret this requirement for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes on the other hand no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.3.3 Specific software requirements (Active electrical energy meters)

Risk Class B	Risk Class C	Risk Class D
I3-1: Fault Recovery *The software shall recover from a disturbance to normal processing.*		
Specifying Notes:		
Required Documentation: A brief description of the fault recovery mechanism and when it is invoked. Brief description of the related tests carried out by the manufacturer.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault recovery is appropriate. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. If any function has not been processed or - in the worst case - the microprocessor hangs in an arbitrary endless loop, the reset of the watchdog does not happen and it fires after a certain time span which resets the microprocessor.		

Risk Class B	Risk Class C	Risk Class D
I3-2: Back-up Facilities *There shall be a facility that provides for the periodic back-up of measurement data, such as measurement values, and the current status of the process. This data shall be stored in non-volatile storage.*		
Specifying Notes: .If the back-up facility is used for fault recovery, the minimum interval shall be calculated to ensure the critical change value is not exceeded.		
Required Documentation: A brief description of which data is backed up and when this occurs.		
Validation Guidance: *Checks based on documentation:* • Check whether measurement data is saved to non-volatile storage and can be recovered. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: Measurement data is backed up as required.		

Risk Class B	Risk Class C	Risk Class D
I3-3: MI-003, 5.2 (indication suitability) *The display of the total energy shall have a sufficient number of digits to ensure that when the meter is operated for 4000 hours at full load ($I = I_{max}$, $U = U_n$ and PF = 1) the indication does not return to its initial value.*		
Specifying Notes:		
Required Documentation: Documentation of the internal representation of the electrical energy register and auxiliary quantities.		
Validation Guidance: *Checks based on documentation:* Check whether number of digits is sufficient (intern and on display).		
Example of an Acceptable Solution: Typical values for three phase electricity meters are: P_{max} (4000h) = 3*60 A * 230 V * 4000h = 165600 kWh. This requires an internal representation of 4 bytes		

Risk Class B	Risk Class C	Risk Class D
I3-4: MID-Annex I, 8.5 (Inhibit resetting of cumulative measurement values) *For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.*		
Specifying Notes: Cumulative registers of a measuring instrument may be reset prior to being put into use.		
Required Documentation: Documentation of protection means against resetting the energy registers.		
Validation Guidance: *Checks based on documentation:* • Check that cumulative legally relevant measurement values cannot be reset without evidence of intervention. *Functional checks:* • Confirm correct functioning. Refer to P3 and P4.		
Example of an Acceptable Solution: The registers for energy are protected against changes and resetting by the same means as parameters (see P7).		

Risk Class B	Risk Class C	Risk Class D
I3-5: Dynamic behaviour *The legally non-relevant software shall not adversely influence the dynamic behaviour of a measuring process.*		
Specifying notes: • This requirement applies in addition to S-1, S-2 and S-3 if software separation has been realised in accordance with extension S. • The additional requirement ensures that for real time applications of meters the dynamic behaviour of the legally relevant software is not inadmissibly influenced by legally non-relevant software, i.e. the resources of the legally relevant software are not inadmissibly reduced by the non-legal part.		
Required Documentation: • Description of the interrupt hierarchy. • Timing diagram of the software tasks. Limits of proportionate runtime for legally non-relevant tasks.		
Validation Guidance: *Checks based on documentation:* • Documentation of the limits of the proportionate runtime for legally non-relevant tasks is available for the programmer of the legally non-relevant software part. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an acceptable solution: The interrupt hierarchy is designed in a way that avoids adverse influences.		

Risk Class B	Risk Class C	Risk Class D
I3-6: Imprinted Software Identifier *The software identifier is usually presented on a display. As an exception for active electrical energy meters, an imprint of the software identifier on the name plate of an instrument shall be an acceptable solution if the following conditions A, B and C are fulfilled:* *A. The user interface does not have any control capability to activate the indication of the software identifier on the display or the display does not allow technically showing the identifier of the software (mechanical counter).* *B. The instrument does not have any interface to communicate the software identifier.* *C. After production of a meter a change of the software is not possible or only possible if also the hardware or a hardware part is changed.*		
Specifying notes: • The manufacturer of the hardware or the concerned hardware part is responsible that the software identifier is correctly marked on the concerned hardware. • All other Specifying Notes of P2/U2 apply.		
Required Documentation: • According to P2/U2.		
Validation Guidance: *Checks based on documentation:* • According to P2/U2. *Functional checks:* • According to P2/U2.		
Example of an acceptable solution: Imprint of the software identifier on the name plate of the instrument.		

10.3.4 Examples of legally relevant parameters, functions, and data

Electronic utility meters often have a lot of parameters. They are used as constants for calculations, as configuration parameters etc but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirement P2 and P7, guide P.

In the following some typical parameters of active electrical energy meters are given. (This table will be updated when WELMEC Working Group 11 has decided on the final contents.)

Parameter	Protected	Settable	Comment
Calibration factor	x		
Linearisation factor	x		

10.3.5 Other aspects

For domestic applications it is expected that download of software (extension D, Chapter 9) will not be very important.

The cumulating energy or volume register of domestic instruments is not a long-term storage in the sense of extension L (Chapter 6). For an instrument that only measures cumulated energy / volume the application of the extension L is not necessary.

10.3.6 Assignment of risk class

For the present, according to the decisions of the responsible WELMEC Working Group 11, the following risk class is considered appropriate and should be applied, if software

examinations based on this guide are carried out for (software-controlled) active electrical energy meters:

- **Risk class C for instruments of type P**

A final decision has, however, not yet been taken and WG 11 will reconsider this item in connection with the discussion of appropriate risk class(es) for type U instruments.

WG 11 considers prepayment and interval metering functionality to be additional to those essential measurement functions specified by MID Annex MI-003, therefore no greater risk category is allocated to these variants than to the basic meter types already covered by this software guide. However, the basic measurement function should be assessed, as with all other type P instruments along with any other assessment deemed necessary to demonstrate that the associated software providing these functions has no inadmissible influence on the basic measurement.

10.4 Thermal Energy Meters

10.4.1 Specific regulations, standards and other normative documents

Member states may – in accordance with MID Article 2 – prescribe Thermal energy meters in residential, commercial and light industrial use to be subject to regulations in MID. The specific requirements of this chapter are based on Annex MI-004 only.

OIML recommendations and standards have not been taken into consideration.

10.4.2 Technical description

10.4.2.1 Hardware Configuration

Thermal energy meters are typically realised as built-for purpose devices (Type P in this document). A heat meter is either a complete instrument or a combined instrument consisting of the sub-assemblies flow sensor, temperature sensor pair, and calculator, as defined in MID Article 4(b), or a combination thereof.

10.4.2.2 Software Configuration

This is specific to each manufacturer but would normally be expected to follow the recommendations given in the main body of this guide.

10.4.2.3 Measuring Principle

Thermal energy meters continually cumulate the energy consumed in a heating circuit. The cumulated thermal energy is displayed at the instrument. Various principles are employed.

The energy measurement may not be repeated.

10.4.2.4 Fault Detection and Reaction

The requirement MI-004, 4.1 and 4.2 deal with electromagnetic disturbances. There is a need to interpret these requirements for software controlled instruments because detection of a disturbance and recovery is only possible by co-operation of specific hardware parts and specific software. From the software point of view it makes no difference what the reason for a disturbance was (electromagnetic, electrical, mechanical etc): the recovery procedures are all the same.

10.4.3 Specific software requirements (Thermal Energy Meters)

Risk Class B	Risk Class C	Risk Class D
I4-1: Fault Recovery *The software shall recover from a disturbance to normal processing.*		
Specifying Notes: Date stamped flags should be raised to help log periods of faulty operation.		
Required Documentation: A brief description of the fault recovery mechanism and when it is invoked.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault recovery is appropriate. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. If any function has not been processed or - in the worst case - the microprocessor hangs in an arbitrary endless loop, the reset of the watchdog does not happen and it fires after a certain time span.		

Risk Class B	Risk Class C	Risk Class D
I4-2: Back-up Facilities *There shall be a facility that provides for periodic back-up of measurement data, such as measurement values and the current status of the process. This data shall be stored in a non-volatile storage.*		
Specifying Notes: The storage intervals shall be sufficiently small so that the discrepancy between the current and saved cumulative values is small.		
Required Documentation: A brief description of which data is backed up and when this occurs. Calculation of the maximum error that can occur for cumulative values.		
Validation Guidance: *Checks based on documentation:* • Check whether measurement data is saved to non-volatile storage and can be recovered. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: Measurement data is backed up as required (e.g. every 60 minutes)		

Risk Class B	Risk Class C	Risk Class D
I4-3: MID-Annex I, 8.5 (Inhibit resetting of cumulative measurement values) *For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.*		
Specifying Notes: Cumulative registers of a measuring instrument may be reset prior to being put into use.		
Required Documentation: Documentation of protection means against resetting the volume registers.		
Validation Guidance: *Checks based on documentation:* • Check that cumulative legally relevant measurement values cannot be reset without leaving a trace. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: The registers for volume are protected against changes and resetting by the same means as parameters (see P7).		

Risk Class B	Risk Class C	Risk Class D
I4-4: Dynamic behaviour *The legally non-relevant software shall not adversely influence the dynamic behaviour of a measuring process.*		
Specifying notes: • This requirement applies in addition to S-1, S-2 and S-3 if software separation has been realised in accordance with extension S. • The additional requirement ensures that for real time applications of meters the dynamic behaviour of the legally relevant software is not inadmissibly influenced by legally non-relevant software, i.e. the resources of the legally relevant software are not inadmissibly reduced by the non-legal part.		
Required Documentation: • Description of the interrupt hierarchy. • Timing diagram of the software tasks. Limits of proportionate runtime for legally non-relevant tasks.		
Validation Guidance: *Checks based on documentation:* • Documentation of the limits of the proportionate runtime for legally non-relevant tasks is available for the programmer of the legally non-relevant software part. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an acceptable solution: The interrupt hierarchy is designed in a way that avoids adverse influences.		

Risk Class B	Risk Class C	Risk Class D
I4-5: Imprinted Software Identifier *The software identifier is usually presented on a display. As an exception for thermal energy meters, an imprint of the software identifier on the name plate of an instrument shall be an acceptable solution if the following conditions A, B and C are fulfilled:* *A. The user interface does not have any control capability to activate the indication of the software identifier on the display or the display does not allow technically showing the identifier of the software (mechanical counter).* *B. The instrument does not have any interface to communicate the software identifier.* *C. After production of a meter a change of the software is not possible or only possible if also the hardware or a hardware part is changed.*		
Specifying notes: • The manufacturer of the hardware or the concerned hardware part is responsible that the software identifier is correctly marked on the concerned hardware. • All other Specifying Notes of P2/U2 apply.		
Required Documentation: • According to P2/U2.		
Validation Guidance: *Checks based on documentation:* • According to P2/U2. *Functional checks:* • According to P2/U2.		
Example of an acceptable solution: Imprint of the software identifier on the name plate of the instrument.		

10.4.4 Examples of legally relevant parameters, functions, and data

Thermal energy meters have parameters like constants for calculations, for configuration etc, but also for setting up the functionality of the device. Concerning identification and protection of parameters and parameter sets refer to requirements P2 and P7, guide P.

In the following some typical parameters of thermal energy meters are given. (This table will be updated when WELMEC Working Group 11 has decided on the final contents.)

Parameter	Protected	Settable	Comment
Calibration factor	x		
Linearisation factor	x		

10.4.5 Other aspects

For domestic applications it is expected that download of software (extension D, Chapter 9) will not be very important.

The cumulating energy or volume register of domestic instruments is not a long-term storage in the sense of extension L (Chapter 6). For an instrument that only measures cumulated energy / volume the application of the extension L is not necessary.

10.4.6 Assignment of risk class

For the present, according to the decisions of the responsible WELMEC Working Group 11, the following risk class is considered appropriate and should be applied, if software examinations based on this guide are carried out for (software-controlled) thermal energy meters:

- **Risk class C for instruments of type P**

A final decision has, however, not yet been taken and WG 11 will reconsider this item in connection with the discussion of appropriate risk class(es) for type U instruments.

10.5 Measuring Systems for the Continuous and Dynamic Measurement of Quantities of Liquids Other than Water

Measuring Systems for the Continuous and Dynamic Measurement of Quantities of Liquids Other than Water are subject to regulations in MID. The specific requirements are in Annex MI-005. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.5.1 – 10.5.2 will be filled in if considered necessary in the future.

10.5.3 Specific software requirements (Measuring System for Liquids other than Water)

Risk Class B	Risk Class C	Risk Class D
I5-1: Imprinted Software Identifier *The software identifier is usually presented on a display. As an exception for measuring systems for liquids other than water, an imprint of the software identifier on the type plate shall be an acceptable solution if the following conditions A, B and C are fulfilled:* *A. The user interface does not have any control capability to activate the indication of the software identifier on the display or the display does not allow technically showing the identifier of the software or there is no display on the instrument.* *B. The instrument does not have any interface to communicate the software identifier.* *C. After production of the instrument a change of the software is not possible or only possible if also the hardware or a hardware part is changed.*		
Specifying notes: • The tag showing the software identifier shall be non-erasable and non-transferable. • The manufacturer of the hardware or the concerned hardware part is responsible that the software identifier is correctly marked on the concerned hardware. • All other Specifying Notes of P2/U2 apply.		
Required Documentation: • According to P2/U2.		
Validation Guidance: *Checks based on documentation:* • According to P2/U2. *Functional checks:* • According to P2/U2.		
Example of an acceptable solution: Imprint of the software identifier on the type plate of the instrument.		

10.5.4 and 10.5.5 will be filled in if considered necessary in the future.

10.5.6 Assignment of risk class

For the present, according to the result of the WELMEC WG 7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) measuring systems for the continuous and dynamic measurement of quantities of liquids other than water.

- **Risk class C**

10.6 Weighing Instruments

Weighing instruments are divided into two main categories:

1. Non-automatic weighing instruments (NAWIs), and
2. Automatic weighing instruments (AWIs).

While most AWIs are governed by the MID, NAWIs are not; they are still governed by the European Directive 90/384/EEC. **Therefore the software guide WELMEC 2.3 applies to NAWIs, whereas this software guide applies to AWIs.**

The specific requirements of this chapter are based on Annex MI-006 and the normative documents mentioned in 10.6.1 as far as they support the interpretation of MID requirements.

10.6.1 Specific regulations, standards and other normative documents

5 categories of automatic weighing instruments (AWIs) are subject to regulations in MID Annex MI-006:

- Automatic catchweighers (R51)
- Automatic gravimetric filling instruments (R61)
- Discontinuous totalisers (R107)
- Continuous totalisers (belt weighers) (R50)
- Automatic rail weighbridges (R106)

The numbers in brackets refer to the respective OIML recommendations that are normative documents in the sense of the MID. In addition, WELMEC has issued the WELMEC Guide 2.6 that supports the testing of automatic catchweighers.

There is one category of AWIs that is not governed by the MID:

- Automatic instruments for weighing road vehicles in motion (R134)

AWIs of all categories may be realised as type P or type U, and all extensions could be relevant for each category.

However, of these 6 categories, only **discontinuous totalisers** and **continuous totalisers** (belt weighers) have been identified as requiring instrument specific software requirements (see 10.6.3). The reason is that the measurement is cumulative over a relatively long period of time and cannot be repeated if a significant fault occurs.

10.6.2 Technical description

10.6.2.1 Hardware Configuration

A discontinuous totaliser is a totalising hopper weigher that determines the mass of a bulk product (e.g. grain) by dividing it into discrete loads. The system usually comprises of one or more hoppers supported on load cells, power supply, electronic controls and indicating device.

A continuous totaliser is a belt weigher that measures the mass of a product as the belt passes over a load cell. The system usually comprises of a conveyor belt, rollers, load receptor supported on load cells, power supply, electronic controls and indicating device. There will be a means for adjusting the tension of the belt.

10.6.2.2 Software Configuration

This is specific to each manufacturer but would normally expect to follow the recommendations given in the main body of this guide.

10.6.2.3 Measuring Principle

In the case of a discontinuous totaliser the bulk product is fed into a hopper and weighed. The mass of each discrete load is determined in sequence and summed. Each discrete load is then delivered to bulk.

In the case of a continuous totaliser the mass is continually measured as the product passes over the load receptor. Measurements are made in discrete units of time that depend on the belt speed and the force on the load receptor. There is no deliberate subdivision of the product or interruption of the conveyor belt as with a discontinuous totaliser. The total mass is an integration of the discrete samples. It should be noted that the load receptor could use strain gauge load cells or other technologies such as vibrating wire.

10.6.2.4 Defects

Joints in the belt may generate shock effects, which can lead to erroneous events when zeroing. In the case of discontinuous totalisers, single or all weighing results of discrete loads may get lost before being summed up.

10.6.3 Specific software requirements (Discontinuous and Continuous Totalisers)

MID Annex MI-006, Chapter IV, Section 8, and Chapter V, Section 6 deals with electromagnetic disturbances. There is a need to interpret these requirements for software controlled instruments because the detection of a disturbance (fault) and subsequent recovery are only possible through the co-operation of specific hardware parts and specific software. From the software point of view, it makes no difference what the reason of a disturbance was (electromagnetic, electrical, mechanical etc); the recovery procedures are all the same.

Risk Class B	Risk Class C	Risk Class D
I6-1: Fault Detection *The software shall detect that normal processing is disturbed.*		
Specifying Notes: On detection of a fault: a. The cumulative measurement and other relevant legal data shall be automatically saved to non-volatile storage (see Requirement I6-2), and b. the hopper weigher or belt weigher shall be stopped automatically, or a visible or audible alarm signal shall be given (see Required Documentation)		
Required Documentation: A brief description of what is checked, what is required to trigger the fault detection process, what action is taken on the detection of a fault. If, on detection of a fault, it is not possible to stop the transportation system automatically without delay (e.g. due to safety reasons) the documentation shall include a description of how the non-measured material is treated or properly taken into account.		
Validation Guidance: *Checks based on documentation:* • Check whether the realisation of fault detection is appropriate. *Functional checks:* • If possible: simulate certain hardware faults and check whether they are detected and reacted upon by the software as described in the documentation.		
Example of an Acceptable Solution: A hardware watchdog is reset by a cyclically processed microprocessor subroutine in order to inhibit the firing of the watchdog. Before resetting, the subroutine checks the health of the system e.g. whether all legally relevant subroutines have been processed during the last interval. If any function has not been processed or - in the worst case - the microprocessor hangs in an arbitrary endless loop, the reset of the watchdog does not happen and it fires after a certain time span.		

Risk Class B	Risk Class C	Risk Class D
I6-2: Back-up Facilities *There shall be a facility that provides for the back-up of measurement data, such as measurement values, and the current status of the process in case of a disturbance.*		
Specifying Notes: a. The state characteristics and important data shall be stored in a non-volatile storage. b. This requirement normally implies a controlled storage facility providing automatic back-up in case of a disturbance. Periodic backing up is acceptable only if a controlled storage facility is not available due to hardware or functional constraints. In that exceptional case the storage intervals shall be sufficiently small, i.e. the maximum possible discrepancy between the current and saved values shall be within a defined fraction of the maximum permissible error (see Required Documentation). c. The back-up facilities should normally include appropriate wake-up facilities in order that the weighing system, including its software, does not get into an indefinite state by a disturbance.		
Required Documentation: A brief description of the back-up mechanism and the data that are backed up, and when this occurs. Specification or calculation of the maximum error that can occur for cumulative values if a cyclical (periodic) back-up is realised.		
Validation Guidance: *Checks based on documentation:* • Check back-up facilities. *Functional checks:* • Check by simulating a disturbance whether back-up mechanism works as described in the documentation.		
Example of an Acceptable Solution: A hardware watchdog fires when it is not cyclically reset. This alarm actuates an interrupt in the microprocessor. The assigned interrupt routine at once collects measurement values, state values and other relevant data and stores them in a non-volatile storage e.g. an EEPROM or other appropriate storage. *Note:* It is assumed that the watchdog interrupt has highest interrupt priority and can dominate any normal processing or any arbitrary endless loop, i.e. the program control always jumps to the interrupt routine if the watchdog fires.		

10.6.4 Examples of legally relevant parameters, functions, and data

Table 10-1: Examples of legally relevant, device-specific and type-specific functions and data (DF, DD, TF, TD) for AWIs in comparison with those of non-automatic weighing instruments (R76). VV indicates variable values.

Functions/data	Type	OIML Recommendation No						
		50	51 (X)	51 (Y)	61	76	106	107
Weight calculation	TF, TD	X	X	X	X	X	X	X
Stability analysis	TF, TD		X	X	X	X	X	X
Price calculation	TF, TD			X		X		
Rounding algorithm for price	TF, TD			X		X		
Span (sensitivity)	DD	X	X	X	X	X	X	X
Corrections for non-linearity	DD (TD)	X	X	X	X	X	X	X
Max, Min, e, d	DD (TD)	X	X	X	X	X	X	X
Units of measurement (e.g. g, kg)	DD (TD)	X	X	X	X	X	X	X
Weight value as displayed (rounded to multiples of e or d)	VV	X		X		X	X	X
Tare, preset tare	VV		X	X	X	X	X	
Unit price, price to pay	VV			X		X		X
Weight value in internal resolution	VV	X	X	X	X	X	X	X
Status signals (e.g. zero indication, stability of equilibrium)	TF	X	X	X	X	X	X	X
Comparison of actual weight vs. preset value	TF		X		X			
Automatic printout release, e.g. at interruption of automatic operation	TF	X						X
Warm-up time	TF (TD)	X	X	X	X	X	X	X
Interlock between functions	TF		X	X				
e.g. zero setting/tare			X	X	X	X		
automatic/non-automatic operation,							X	
zero-setting/totalizing		X						X
Record of access to dynamic setting	TF (VV)		X	X				
Maximum rate of operation/range of operating speeds (dynamic weighing)	DD (TD)	X	X	X	X		X	X
(Product)-Parameters for dynamic weight calculation	VV		X	X			X	
Preset weight value	VV		X		X			
Width of adjustment range	DD (TD)		X	X				
Criterion for automatic zero-setting (e.g. time interval, end of weighing cycle)	DD (TD)		X	X	X		X	X
Minimum discharge, rated minimum fill	DD				X			X
Limiting value of significant fault (if not 1e or 1d)	DD (TD)	X			X			
Limiting value of battery power	DD (TD)	X	X	X	X	X	X	X

Table 10-1: Examples of legally relevant, device-specific and type-specific functions and data

The marked functions and parameters are likely to occur on the various types of weighing instruments. If one of them is present, it has then to be treated as “legally relevant”. The table is, however, not meant as an obligatory list indicating that any function or parameter mentioned has to be realised in each instrument.

10.6.5 Other aspects

None

10.6.6 Assignment of risk class

For the present, according to the decision of the responsible WELMEC Working Group (24th WG 2 meeting, 22/23 January 2004) **risk class "B" shall be generally applied** to all categories of AWIs regardless of the type (P or U).

However, as a result of the WG 7 questionnaire (2004), the following differentiation with regard to type P and U instruments, and to discontinuous and continuous totalising instruments (="totalisers") seems appropriate:

- **Risk class B for type P instruments (except totalisers)**
- **Risk class C for type U instruments and totalisers type P and U**

10.7 Taximeters

Taximeters are subject to regulations in MID. The specific requirements are in Annex MI-007. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.7.1 Specific regulations, standards and normative documents

The European Standard EN50148 which could become a normative documents in the sense of the MID has not been yet considered. There is a publication of a guidance document about taximeters as a result of the MID-Procedures project. In future this document will be the basis of a WELMEC Guide. Also there is a very first draft of an OIML Recommendation on taximeters. The OIML document is however not in a stage where it could be used as a normative document (situation of October 2004).

10.7.2 Technical description

A taximeter as defined in MID measures the time, the distance (using the output of a distance signal generator not covered by MID) and calculates the fare for a trip based on the applicable tariffs.

Current taximeters use an embedded architecture, which means taximeters are built-for-purpose instruments (type P) in the sense of this guide. In future it is expected that taximeters will also be manufactured using universal computers (type U).

10.7.3 Specific software requirements

MID Annex MI-007, 9:

> In case of a reduction of the voltage supply to a value below the lower operating limit as specified by the manufacturer, the taximeter shall:
>
> – continue to work correctly or resume its correct functioning without loss of data available before the voltage drop if the voltage drop is temporary, i.e. due to restarting the engine,
> – abort an existing measurement and return to the position "For Hire" if the voltage drop is for a longer period.

The taximeter also needs to have a long-term storage, the data shall be available in the taximeter for at least 1 year, see MI-007, 15.2.

Risk Class B	Risk Class C	Risk Class D
I7-1: Back-up Facilities *There shall be a facility that automatically backs-up essential data, e.g. measurement values and the current status of the process if the voltage drops for a longer period.*		
Specifying Notes: 1) This data should normally be stored in non-volatile storage. 2) A voltage level detector to detect when to store measurement values is necessary. 3) The back-up facilities shall include appropriate wake-up facilities in order that the taximeter, including its software, does not get into an indefinite state.		
Required Documentation: A brief description of which data is backed up and when this occurs.		
Validation Guidance: *Checks based on documentation:* • Check whether measurement data is saved in case of a disturbance.. *Functional checks:* • Confirm correct functioning in the presence of defined influencing quantities and provoked errors.		
Example of an Acceptable Solution: The voltage level detector fires an interrupt when the voltage level drops for a time of 15 s. The assigned interrupt routine collects measurement values, state values, and other relevant data and stores them in a non-volatile storage e.g. EEPROM. After the voltage level rises again the data is restored and the functioning continues or is stopped (see MI-007, 9.) *Note:* It is assumed that the voltage level interrupt has a high interrupt priority and can dominate any normal processing or any arbitrary endless loop, i.e. the program control always jumps to the interrupt routine if the voltage drops.		

10.7.4 Examples of legally relevant parameters, functions, and data

In the following some typical parameters of taximeters are given.

Parameter	Protected	Settable	Comment
k-factor	x		Impulses per km
Tariffs	x	x	Currency Unit/km, Currency Unit/h
Interface parameters		x	Baud-rate etc

10.7.5 Other aspects

It is recommended that the Automotive Directive is revised or any other regulation is made to give requirements for the distance signal generators of vehicles used as taxi. A preliminary proposal reads:

For vehicles intended to be used as taxi the following requirements apply:

1. The distance signal generator shall give a signal with a resolution of at least 2 m.
2. The distance signal generator shall give a stable signal at every speed travelled.
3. The distance signal generator shall have defined characteristics regarding voltage level, pulse width and the relation of speed and frequency.
4. Testability…

10.7.6 Assignment of risk class

For the present, according to the result of the WELMEC WG 7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) taximeters:

- **Risk class C for type P instruments**
- **Risk class D for type U instruments**

10.8 Material Measures

Material measures are subject to regulations in MID. The specific requirements are in Annex MI-008.

Subject to future developments and decisions material measures in the sense of MID Annex MI-008 are not considered to be software-controlled measuring instruments. Thus, for the present, this software guide does not apply to material measures.

10.9 Dimensional Measuring Instruments

Dimensional Measuring Instruments are subject to regulations in MID. The specific requirements are in Annex MI-009. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.9.1 - 10.9.5 will be filled in if considered necessary in the future.

10.9.6 Assignment of risk class

For the present, according to the result of the WELMEC WG 7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) dimensional measuring instruments:

- **Risk class B for type P instruments**
- **Risk class C for type U instruments**

10.10 Exhaust Gas Analysers

Exhaust Gas Analysers are subject to regulations in MID. The specific requirements are in Annex MI-010. Neither these specific requirements nor any normative documents have yet been taken into consideration.

10.10.1 - 10.10.5 will be filled in if considered necessary in the future.

10.10.6 Assignment of risk class

For the present, according to the result of the WELMEC WG 7 questionnaire (2004) and subject to future decisions of the responsible WELMEC Working Group, the following risk class should be applied if software examinations based on this guide are carried out for (software-controlled) exhaust gas analysers:

- **Risk class B for type P instruments**
- **Risk class C for type U instruments**

11 Pattern for Test Report (Including Checklists)

This is a pattern for a test report, which consists of a main part and two annexes. The main part contains general statements on the object under test. It must be correspondingly adapted in practice. The annex 1 consists of two checklists to support the selection of the appropriate parts of the guide to be applied. The annex 2 consists of specific checklists for the respective technical parts of the guide. They are recommended as an aid for manufacturer and examiner to prove that they have considered all applicable requirements.

In addition to the pattern of the test report and the checklists, the information required for the type examination certificate is listed in the last subsection of this chapter.

11.1 Information to be included in the type examination certificate

While the entire test report is a documentation of the object under test, the validation carried out and the results, a certain selection of the information contained in the test report are required for type examination certificate (TEC). This concerns the following information, which should be appropriately included in the TEC:

- Reference to the documentation submitted for type examination,
- Identification and description of the electronic (hardware) parts (subassemblies, modules) that are important for software/IT function of the measuring instruments,
- Overview of the software environment, which is necessary to operate the software,
- Overview of SW modules under legal control (including SW separation, if implemented),
- Overview and identification of hardware and software (if relevant) interfaces that are important for software / IT functions of the measuring instruments (including infrared, Bluetooth, Wireless LAN, …),
- Identification and description of locations of software parts in the measuring instrument (i.e. EPROM, processor, hard disk, …) that need to be sealed or secured,
- Instructions of how to check the identification of software (for metrological supervision),
- Procedure or means for the integrity check of the legal relevant software (mechanical sealing, checksum, event logger, event counter, etc.) should be stated. When relevant they should be related to specific software version in TEC,
- Instruction for the inspection of event counter / event logger.

11.2 Pattern for the general part of the test report

Test report no XYZ122344

Flow meter Dynaflow model DF101

Validation of Software

(n annexes)

Commission

The Measuring Instruments Directive (MID) gives the essential requirements for certain measuring instruments used in the European Union. The software of the measuring instrument was validated to show conformance with the essential requirements of the MID.

The validation was based on the report WELMEC MID Software Requirements Guide WELMEC Guide 7.2, where the essential requirements are interpreted and explained for software. This report describes the examination of software needed to state conformance with the MID.

Client

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

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

According to the WELMEC Guide 7.2, the flow meter is described as follows:

- a built-for-purpose Measuring instrument (an embedded system)
- long-term storage of measurement data

The flow meter DF100 is an independent instrument with a transducer connected. The transducer 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 version of the software validated is **V1.2c**. The source code comprises following 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 validation has been supported by following documents from the manufacturer:

- DF 100 User Manual
- DF 100 Maintenance Manual
- 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.

Examination Procedure

The validation has been performed according to the WELMEC 7.2 Software Guide, Issue 1 (downloaded at www.welmec.org).

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:

- Specific requirements for embedded software for a built-for-purpose measuring instrument (type P)
- Extension L: Long-term storage for measurement data

Checklist for the selection of the configuration is found in annex 1 to this report.

Risk class C has been applied to this instrument.

Following validation methods have been applied:

- identification of the software
- completeness of the documentation
- examination of the operating manual
- functional testing
- software design review
- review of software documentation
- data flow analysis
- simulation of input signals

Result

Following requirements of the WELMEC Software Guide 7.2 have been validated without finding faults:

- P1, P2, P3, P5, P6, P7
  (Requirement P4 is considered to be non-applicable.)
- L1, L2, L3, L4, L5, L6, L7

Checklists for the P-requirements are found in annex 2.1 of this report.

Checklists for the L-requirements are found in annex 2.2 of this report.

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 software of the Dynaflow DF100 V1.2c fulfils the essential requirements of the Measuring Instruments Directive.

The result applies to the tested item only.

National Testing & Measurement Lab
Software Department

Dr. K.E.I.N. Fehler
Technical manager

M. S.A.N.S Problème
Technical Officer

Date: 23 December 2003

Page 3 / 3

11.3 Annex 1 of the test report: Checklists to support the selection of the appropriate requirement Sets

The first checklist supports the user to decide which of basic configuration P or U applies for the instrument under test.

Decision on Instrument Type				
		(P)		**Remarks**
1	Is the entire application software constructed for the measuring purpose?			
		(Y)		
2	Are the requirements for the inclusion of an operating system or subsystems of it fulfilled?			
		(Y)		
3	Is the user prevented from accessing the operating system if it is possible to switch to an operating mode not subject to legal control?			
		(Y)		
4	Are the implemented programs and the software environment invariable (apart from updates)?			
		(Y)		
5	Are there any means for programming?			
		(N)		
Tick the empty boxes, as appropriate				

If and only if all answers to the 5 questions can be given as in the (P) column, then the requirements of the part P (Chapter 4) apply. In all other cases the requirements of the part U (Chapter 50) are necessarily to apply.

The second checklist supports to decide which of the IT configuration applies for the instrument under test.

Decision on Required Extensions					
Req. Extension		**YES**	**NO**	**Not Applicable**	**Remarks**
L	Does the device have the ability to store the measurement data either on an integrated storage or on a storage of universal computer or on a remote or removable storage?				
T	Is measurement data transmitted via communication networks to a distant device where it is further processed and/or used for legally relevant purposes?				
S	Are there software parts with functions not subject to legal control AND are these software parts desired to be changed after type approval?				
D	Is loading of software possible or desired after putting the measuring instrument into use?				
Consider the required extension for each question answered with YES!					

11.4 Annex 2 of the test report: Specific checklists for the respective technical parts

1) Checklist of basic requirements for type P instrument

Checklist for Type P Requirements						
Requirement	**Testing procedures**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
P1		Does the required manufacturer documentation fulfil the requirement P1 (a-f)?				
P2		Is a software identification realised as required in P2?				
P3		Are commands entered via the user interface prevented from inadmissibly influencing the legally relevant software and measurement data?				
P4		Do commands inputted via communication interfaces of the instrument not inadmissibly influence the legally relevant software, device-specific parameters and measurement data?				
P5		Are legally relevant software and measurement data protected against accidental or unintentional changes?				
P6		Is the legally relevant software secured against the inadmissible modification, loading or swapping of hardware memory?				
P7		Are legally relevant parameters secured against unauthorised modification?				
** Explanations are needed if there are deviations from software requirements.*						

2) Checklist for basic requirements for type U instrument

Checklist for Type U Requirements						
Requirement	**Testing procedures**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
U1		Does the required manufacturer's documentation fulfil the requirement U1 (a-g)?				
U2		Is a software identification realised as required in U2?				
U3		Are commands entered via the user interface prevented from inadmissibly influencing the legally relevant software and measurement data?				
U4		Do commands inputted via communication interfaces of the device not inadmissibly influence the legally relevant software, device-specific parameters and measurement data?				
U5		Are legally relevant software and measurement data protected against accidental or unintentional changes?				
U6		Are legally relevant software and measurement data secured against intended, inadmissible modification or replacement?				
U7		Are legally relevant parameters secured against unauthorised modification?				
U8		Is the authenticity of the measurement data that are presented guaranteed?				
U9		Is the legally relevant software designed in such a way that other software does not inadmissibly influence it?				
** Explanations are needed if there are deviations from software requirements.*						

3) Checklist for specific requirements extension L

Checklist for Requirements of Extension L						
Requirement	**Testing procedures**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
L1		Is the stored measurement data accompanied by all relevant information needed for legally relevant purposes?				
L2		Is stored data protected against accidental and unintentional changes?				
L3		Is the stored measurement data protected against intentional changes?				
L4		Is the stored measurement data capable of being authentically traced back to the measurement that generated them?				
L5		Are keys and associated information treated as measurement data and are they kept secret and protected against compromise?				
L6		Is there legally relevant software for displaying or printing stored measurement data?				
L7		Is the measurement data stored automatically when the measurement is concluded?				
L8		Does the long-term storage have a capacity which is sufficient for the intended purpose?				
** Explanations are needed if there are deviations from software requirements.*						

4) Checklist for specific requirements extension T

Checklist for Requirements of Extension T						
Requirement	**Testing procedures**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
T1		Does transmitted data contain all relevant information necessary to present or further process the measurement result in the receiving unit?				
T2		Is transmitted data protected against accidental and unintentional changes?				
T3		Is legally relevant transmitted data protected against intentional changes?				
T4		Is the authenticity of transmitted measurement data ensured?				
T5		Are keys and associated information treated as measurement data and kept secret and protected against compromise?				
T6		Is data that is detected as having been corrupted marked_to enable further processing software to react accordingly?				
T7		Is it ensured that the measurement is not inadmissibly influenced by a transmission delay?				
T8		Is it ensured that no measurement data get lost if network services become unavailable?				
** Explanations are needed if there are deviations from software requirements.*						

5) Checklist for specific requirements extension S

Checklist for Requirements of Extension S						
Requirement	**Testing procedures**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
S1		Is there a part of the software that contains all legally relevant software and parameters that is clearly separated from other parts of software?				
S2		Is information generated by the legally non-relevant software shown on a display or printout in a way that confusion with the information generated by the legally relevant software is avoided?				
S3		Is the data exchange between the legally relevant and legally non-relevant software carried out exclusively via a protective software interface?				
** Explanations are needed if there are deviations from software requirements.*						

6) Checklist for specific requirements extension D

Checklist for Requirements of Extension D						
Requirement	**Testing procedures**		**Passed**	**Failed**	**Not Applicable**	**Remarks***
D1		Do both phases of the software download, the transmission, and the subsequent installation of software, run automatically and do they not affect the protection of legally relevant software?				
D2		Are means employed to guarantee that the downloaded software is authentic?				
D3		Are means employed to guarantee that the downloaded software has not been inadmissibly changed during download?				
D4		Is it guaranteed by appropriate technical means that downloads of legally relevant software are adequately traceable within the instrument for subsequent controls?				
** Explanations are needed if there are deviations from software requirements.*						

12 Cross Reference for MID-Software Requirements to MID Articles and Annexes

(Related MID Version: DIRECTIVE 2014/32/EU, 26 February 2014)

12.1 Given software requirement, reference to MID

Requirement		MID	
No	**Denotation**	**Article / Annex No** (AI = Annex I)	**Denotation**
	Basic Guide P		
P1	Manufacturer's Documentation	AI-9.3 AI-12 Article 18	Information to be borne by and to accompany the instrument Conformity Evaluation Technical Documentation
P2	Software Identification	AI-7.6 AI-8.3	Suitability Protection against corruption
P3	Influence via User Interface	AI-7.1	Suitability
P4	Influence via communication Interface	AI-7.1 AI-8.1	Suitability Protection against corruption
P5	Protection Against Accidental or Unintentional Changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
P6	Protection Against Intentional Changes	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability[2] Protection against corruption
P7	Parameter Protection	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability Protection against corruption
	Basic Guide U		
U1	Manufacturer's Documentation	AI-9.3 AI-12 Article 18	Information to be borne by and to accompany the instrument Conformity Evaluation Technical Documentation

[2] Note: As regards contents, paragraph 7.1 of MID-Annex I is not an issue of "Suitability" but of "Protection against corruption" (Paragraph 8)

Requirement		MID	
No	**Denotation**	**Article / Annex No** (AI = Annex I)	**Denotation**
U2	Software Identification	AI-7.6 AI-8.3	Suitability Protection against corruption
U3	Influence via user interfaces	AI-7.1	Suitability
U4	Influence via Communication Interface	AI-7.1 AI-8.1	Suitability Protection against corruption
U5	Protection against accidental or unintentional changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
U6	Protection against Intentional Changes	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability Protection against corruption
U7	Parameter Protection	AI-7.1 AI-8.2, AI-8.3, AI-8.4	Suitability Protection against corruption
U8	Software authenticity and Presentation of Results	AI-7.1, AI-7.2, AI-7.6 AI-8.3 AI-10.2, AI-10.3, AI-10.4	Suitability Protection against corruption Indication of result
U9	Influence of other software	AI-7.6	Suitability
	Extension L		
L1	Completeness of stored data	AI-7.1 AI-8.4 AI-10.2	Suitability Protection against corruption Indication of result
L2	Protection against accidental or unintentional changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
L3	Integrity of data	AI-7.1 AI-8.4	Suitability Protection against corruption
L4	Authenticity of stored data	AI-7.1 AI-8.4 AI-10.2	Suitability Protection against corruption Indication of result
L5	Confidentiality of keys	AI-7.1 AI-8.4	Suitability Protection against corruption
L6	Retrieval of stored data	AI-7.2 AI-10.1, AI-10.2, AI-10.3, AI-10.4	Suitability Indication of result
L7	Automatic storing	AI-7.1 AI-8.4	Suitability Protection against corruption
L8	Storage capacity and continuity	AI-7.1	Suitability
Lx	All of Extension L	AI-11.1	Further processing of data to conclude the trading transaction
	Extension T		
T1	Completeness of transmitted data	AI-7.1 AI-8.4	Suitability Protection against corruption
T2	Protection against accidental changes	AI-7.1, AI-7.2 AI-8.4	Suitability Protection against corruption
T3	Integrity of data	AI-7.1 AI-8.4	Suitability Protection against corruption
T4	Authenticity of transmitted data	AI-7.1 AI-8.4	Suitability Protection against corruption
T5	Confidentiality of keys	AI-7.1 AI-8.4	Suitability Protection against corruption
T6	Handling of corrupted data	AI-7.1 AI-8.4	Suitability Protection against corruption
T7	Transmission delay	AI-7.1 AI-8.4	Suitability Protection against corruption
T8	Availability of transmission services	AI-7.1 AI-8.4	Suitability Protection against corruption
	Extension S		
S1	Realisation of software separation	AI-7.6, AI-10.1	Suitability Indication of result
S2	Mixed indication	AI-7.1, AI-7.2, AI-7.6 AI-10.2	Suitability Indication of result
S3	Protective software interface	AI-7.6	Suitability
	Extension D		
D1	Download mechanism	AI-8.2, AI-8.4	Protection against corruption

Requirement		MID	
No	**Denotation**	**Article / Annex No** (AI = Annex I)	**Denotation**
D2	Authentication of downloaded software	AI-7.6 AI-8.3, AI-8.4 AI-12	Suitability Protection against corruption Conformity evaluation
D3	Integrity of downloaded software	AI-7.1, AI-8.4	Suitability Protection against corruption
D4	Traceability of legally relevant Software Download	AI-7.1, AI-7.6 AI-8.2, AI-8.3 AI-12	Suitability Protection against corruption Conformity evaluation
	Extension I (Instrument specific Software Requirements)		
I1-1, I2-1, I3-1, I4-1	Fault Recovery	AI-6 MI-001-7.1, MI-002-3.1, MI-003-4.3.1, MI-004-4	Reliability Specific Requirements for Utility Meters
I1-2, I2-2, I3-2, I4-2	Back-up facilities	AI-6 MI-001-7.1, MI-002-3.1, MI-003-4.3.1, MI-004-4	Reliability Specific Requirements for Utility Meters
I1-4, I2-4, I3-4, I4-4	Internal resolution, suitability of the indication	MI-002-5.3, MI-003-5.2	Specific Requirements for Utility Meters
I1-3, I2-4, I3-4, I4-3	Inhibit resetting of cumulative measurement values	AI-8.5	Protection against corruption
I1-4, I2-8, I3-5, I4-4	Dynamic behaviour	AI-7.6	Suitability Protection against corruption
I2-5	Battery lifetime	MI-002-5.2	Specific Requirements for Gas Meters
I2-6	Electronic volume converters	MI-002-9.1	Specific Requirements for Gas Meters
I2-7	Test element	MI-002-5.5	Specific Requirements for Gas Meters
I6-1	Fault detection	MI-006-IV, MI-006-V	Discontinuous and continuous Totalisers
I6-2	Back-up facilities Fault detection	MI-006-IV, MI-006-V	Discontinuous and continuous Totalisers

12.2 Interpretation of MID Articles and Annexes by MID-Software Requirements

MID			Software Guide
Article / Annex No (AI = Annex I)	**Denotation**	**Comment**	**Requirement No**
	Article Part		
1, 2, 3		No specific software relevance	
4(b)	Definitions, Arrangement of sub-assemblies	Transmission of measurement data ... Basic Guides applicable to sub-assemblies	T P, U
5 to 9		No specific software relevance	
10	Technical documentation	Documentation of design, manufacture and operation. Enable assessment of conformity. General description of the instrument. Description of electronic devices with drawings, flow diagrams of the logic, general software information. Location of seals and markings. Conditions for compatibility with interfaces and sub-assemblies.	P1, U1

MID			Software Guide
Article / Annex No (AI = Annex I)	**Denotation**	**Comment**	**Requirement No**
11 to 27		No specific software relevance	
	Annex I		
AI-1 to AI-5		No specific software relevance	
AI-6	Reliability	Fault detection, back-up, restoring, restart	I1-1, I1-2~~3~~, I2-1, I2-2~~3~~, I3-1, I3-2~~3~~, I4-1, I4-2~~3~~, I6-1, I6-2
AI-7	Suitability	No features to facilitate fraudulent use; minimal possibilities for unintentional misuse.	P3 - P7, U3 - U8, L1 – L5, L7, L8, T1 – T8, S2, D3, D4, I1-4, I2-8, I3-5, I4-4
AI-8	Protection against corruption		
AI-8.1		No influences by the connection of other devices.	P4, U4
AI-8.2		Securing; evidence of intervention	P6, P7, U6, U7, D1, D4
AI-8.3		Identification of software; evidence of intervention	P2, P6, P7, U2, U6, U7, U8, D2, D4
AI-8.4		Protection of stored or transmitted data	P5 - P7, U5 - U7, L1 - L5, T1 - T8 D1 - D3
AI-8.5		No reset of cumulative registers	I1-3, I2-4, I3-4, I4-3
AI-9	Information to be borne by and to accompany the instrument		
AI-9.1		Measuring capacity (rest of items non-relevant for software)	L8
AI-9.2		No specific software relevance	
AI-9.3		Instructions for installation, ..., conditions for compatibility with interface, sub-assemblies or measuring instruments.	P1, U1
AI-9.4 to AI-9.8		No specific software relevance	
AI-10	Indication of result		
AI-10.1		Indication by means of a display or hard copy.	U8, L6, S2
AI-10.2		Significance of result, no confusion with additional indications.	U8, L1, L4, L6, S2
AI-10.3		Print or record easily legible and non-erasable.	U8, L6, S2
AI-10.4		For direct sales: presentation of the result to both parties.	U8, S2
AI-10.5		For utility meters: display for the customer.	I1-3, I2-3, I3-3/4, I4-3
AI-11	Further processing of data to conclude the trading transaction		
AI-11.1		Record of measurement results by a durable means.	L1 - L8
AI-11.2		Durable proof of the measurement result and information to identify a transaction.	L1, L6
AI-12	Conformity evaluation	Ready evaluation of the conformity with the requirements of the Directive.	P1, P2, U1, U2, D2, D4
	Annexes A1 to H1		

MID			Software Guide
Article / Annex No (AI = Annex I)	**Denotation**	**Comment**	**Requirement No**
A1 to H1		No requirements to features of instruments	
	Annex MI-001		
MI-001-1 to MI-001-6		No specific software relevance	
MI-001-7.1.1, MI-001-7.1.2	Electromagnetic immunity	Fault detection Back-up facilities Wake-up facilities and restoring	I1-1, I1-2
MI-001-7.1.3 to MI-001-9		No specific software relevance	
	Annex MI-002		
MI-002-1 to MI-002-2		No specific software relevance	
MI-002-3.1	Electromagnetic immunity	Fault detection Back-up facilities Wake-up facilities and restoring	I2-1, I2-2
MI-002-3.1.3 to MI-002-5.1		No specific software relevance	
MI-002-5.2	Suitability	Acceptable solution for monitoring battery lifetime	I2-5
MI-002-5.3	Suitability	Internal resolution	I2-3
MI-002-5.4 to MI-002-8		No specific software relevance	
MI-002-5.5	Suitability	Test element	I2-7
MI-002-5.6 to MI-002-8		No specific software relevance	
MI-002-9.1	Volume conversion devices Suitability	Acceptable solution for monitoring the gas volume converter	I2-6
MI-002-9.2 to MI-002-10		No specific software relevance	
	Annex MI-003		
MI-003-1 to MI-003-4.2		No specific software relevance	
MI-003-4.3	Permissible effect of transient electromagnetic phenomena	Fault detection Back-up facilities Wake-up facilities and restoring	I3-1, I3-2
MI-003-5.1		No specific software relevance	
MI-003-5.2	Suitability	Internal resolution	I3-3
MI-003-5.3 to MI-003-7		No specific software relevance	
	Annex MI-004		
MI-004-1 to MI-004-4.1		No specific software relevance	
MI-004-4.2	Permissible influences of electromagnetic disturbances	Fault detection Back-up facilities Wake-up facilities and restoring	I4-1, I4-2
MI-004-4.3 to MI-004-7		No specific software relevance	
	Annex MI-005		
	Annex MI-006		
MI-006-IV, MI-006-V	Discontinuous and continuous Totalisers	Fault detection Back-up facilities	I6-1, I6-2
	Annex MI-007		

MID			Software Guide
Article / Annex No (AI = Annex I)	**Denotation**	**Comment**	**Requirement No**
MI-007-8	Permissible influences of electromagnetic disturbances	Back-up facilities	I7-1
	Annex MI-008		
	Annex MI-009		
	Annex MI-010		

13 References and Literature

[1] Directive 2004/22/EC of the European Parliament and of the Council of 31 March 2004 on measuring instruments. Official Journal of the European Union L 135/1, 30.4.2004

[2] Software Requirements and Validation Guide, Version 1.00, 29 October 2004, European Growth Network "*MID-Software",* contract number G7RT-CT-2001-05064, 2004

[3] Software Requirements on the Basis of the Measuring Instruments Directive, WEMEC 7.1, Issue 2, 2005

[4] Internet Security Glossary, http://www.ietf.org/rfc/rfc2828.txt

[5] ISO/IEC JTC1/SC7 3941, 2008-03-14, http://pef.czu.cz/~papik/doc/MHJS/pdf/IT-VOCABULARY.pdf

[6] http://www.oxforddictionaries.com/definition/english/audit-trail

[7] DIRECTIVE 2014/32/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 26 February 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of measuring instruments (recast), Official Journal of the European Union L 96/149, 29.3.2014

14 Revision History

Issue/version	Significant Changes
1	Guide first issued.
2	Addition and enhancement of terms in Section 2 Editorial changes in Sections 4.1 and 5.1 Amendment of a clarification for software identification in Section 4.2, Requirement P2 and Section 5.2, Requirement U2. Amendment in Requirement L8, Specifying Note 1. Addition of an explanation to Requirement S1, Specifying Note 1. Replacement of Requirement D5 by a remark. Change of the Risk Class for Measuring Systems for Liquids other than Water. Change of Risk Classes for Weighing Instruments. Various minor editorial changes in the document. Addition of this revision table.
3	Addition of exceptions for the indication of the software identification: new requirements I1-5, I2-9, I3-6, I4-5, and I5-1.
4	Restriction of the application area of software download, clarification of identification requirements in connection with software download Revision of requirements P2 and U2: Deletion of void text fragments.
5	Revision of chapter 5 (part U): Advancement with respect to operating systems Replacement of the term "component" by other appropriate terms through the guide to avoid misunderstandings Addition of requirement D1 in section 9.2 by introduction of a sealable setting for

	the download mechanism Refinement of the specifying notes of requirements P2 and U2 in section 4.2 and 5.2, respectively, with regard to software identification Extension of examples of acceptable solutions in requirement L2 (section 6.2) and in requirement U8 (section 5.2)
2015	Major revision - Character of the guide: The guide is considered a purely technical document that interprets software-related essential requirements. Statements that do not correspond to this principle have been removed. - Addressees of the guide: The guide addresses software developers and examiners, but may be used as well by other parties, in particular Market Surveillance Authorities, wherever and whenever it is appropriate. - It has turned out that the implementation of the two latter updates requires much editorial work in detail. These changes will lead to a better readability of the guide, but not change technical specifications. - Software identification (P2/U2): It shall not be anymore required in the guide 7.2 that the software identifier has to be provided by the software itself. It is sufficient to require that the software identifier has to be provided by the instrument in a secured way. - Differentiation between identification and integrity (P2/U2, P6/U6): MID annex 1 distinguishes between identification of software (annex 1, cl. 7.6) and integrity, e.g. protection of software (annex 1, cl. 8.4). The differentiation does not lead to weaker requirements. - Support of conformity-to-type checks: The technical means required for integrity of software are considered suitable also to be used for the check of conformity to type. The means required are e.g. checksums or equivalent means at different levels for all instruments in risk class C and higher. - Risk classes: Risk class C has been changed so that now the whole legally relevant software is considered fixed for instruments in risk class C. In this way, ambiguities which part of software is considered fixed have been removed. In risk class C and higher identity of software on the bit level (e.g. by checksums) must be implemented. - Risk classification of instruments with universal computers (U type instruments): Due to a basically higher risk associated with U type instruments, their classification into risk class B is considered inappropriate. U type instruments can only be classified into risk class C upwards. - Acceptable security measures for high Risk Classes (D and higher): Concerning algorithms and minimum key lengths, the requirements or recommendations of the national and international institutions responsible for data security have to be taken into consideration (e.g. NIST (USA), DCSSI (France), CESG (United Kingdom), CCN (Spain), NCSC (Netherlands), BSI (Germany)). - Legally relevant software: It is not seen anymore the necessity to differentiate between legally relevant software and fixed legally relevant software. All protection requirements in annex I are valid for legally relevant software.

Table 14-1: Revision history