

WELMEC 2.3

# WELMEC

European cooperation in legal metrology

## **Guide for Examining Software (Non-automatic Weighing Instruments)**

January 1995

## Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
1.1	Background .....	3
1.2	General considerations .....	4
1.3	Scope .....	4
1.4	Conception .....	6
<b>2</b>	<b>Terminology .....</b>	<b>7</b>
2.1	Legally relevant software .....	7
2.2	Changes of software .....	9
2.3	Protection of software .....	9
<b>3</b>	<b>Software requirements .....</b>	<b>9</b>
3.1	Protection of the legally relevant software .....	9
3.2	Software interfaces .....	11
3.3	Software identification .....	12
3.4	Documentation .....	12
<b>4</b>	<b>Report on the software examination .....</b>	<b>13</b>
<b>5</b>	<b>Required specifications in certificates .....</b>	<b>13</b>
<b>6</b>	<b>Preliminary procedure .....</b>	<b>13</b>

# 1 Introduction

## 1.1 Background

The EC Directive 90/384/EEC states some essential requirements for the protection against changes, manipulation or fraudulent use of non automatic weighing instruments (NAWIs) which, in principle, have to be applied also to the software controlling these instruments:

- (i) Annex 1, No 8.1, Directive 90/384/EEC:  
Design and construction of the instruments shall be such that the instruments will preserve their metrological qualities when properly used and installed, and when used in an environment for which they are intended .....
- (ii) Annex 1, No 8.5, Directive 90/384/EEC:  
The instruments shall have no characteristics likely to facilitate fraudulent use, whereas possibilities for unintentional misuse shall be minimal. Components that may not be dismantled or adjusted by the user shall be secured against such actions.
- (iii) Annex II, No 1.7, Directive 90/384/EEC:  
The applicant shall keep the notified body that has issued the EC type-approval certificate informed of any modification to the approved type .....

In the practice of type examination of NAWIs by the various European Notified bodies, it has become apparent that the above essential requirements very urgently need a uniform interpretation with regard to intelligent user-accessible (freeprogrammable) peripheral devices or modules of NAWIs, such as PC-based indicators, data storage devices or point of sale devices (POS).

The item of 'Software Requirements for NAWIs was raised at the 7th WG2 meeting on 23 February 1994 where a respective discussion paper and questionnaire of the PTB was circulated. The results were discussed at the 8th WG2 meeting at SP in Borås on 6/7 June 1994, where it was decided to cooperate with CECIP (the European Committee of manufacturers of weighing instruments) on this matter.

A WG2 subgroup was constituted, consisting of CECIP, DADTI (Denmark), DELTA (Denmark), NMI (The Netherlands), NWML (UK), PTB (Germany), SdM (France) and SP (Sweden).

On the occasion of a subgroup meeting in Berlin in 5/6 September 1994, a consensus of all participants - including CECIP - was achieved about a '5 point catalogue' of software requirements for freeprogrammable, PC-based modules or peripheral devices which are linked to, or form part of NAWIs subject to legal control.

On the basis of this catalogue, a draft of 'Requirements on software for NAWIs subject to legal control' was worked out which was circulated among all subgroup members and finally discussed and its principles agreed upon at the 9th WELMEC WG2 meeting in Brussels, 22/24 November 1994. Both sides, the representatives of the Notified Bodies responsible for type examinations of NAWIs and the representatives of CECIP fully agreed that there is a very urgent need for a 'Guide for examining software for non-automatic weighing instruments (NAWI)', and

that such a document should be issued as soon as possible in order to gain experience and knowledge with the approach presented hereafter.

## 1.2 General considerations

The European Standard on non-automatic weighing instruments, EN 45501, specifies the metrological and technical requirements for non-automatic weighing instruments subject to legal metrological control in order to meet the essential requirements of EC Directive 90/384/EEC. The requirements of this European Standard apply to all devices performing the relevant functions, whether integrated in an instrument or manufactured as a separate unit (see EN 45501, point 2.4).

A problem with the software of weighing instruments, modules or peripheral devices is that this standard does not describe the relevant requirements and examinations to be applied to the software of these instruments or modules and how the result of the examination is to be documented.

This guide tries to fill this gap with regard to software for free programmable, PC-based devices which are linked to, or form part of, NAWIs subject to legal control.

The basic intention of this Guide is to:

- *Describe essential properties* of the software rather than technical solutions.
- *Offer an effective*, but not an *extensive protection against manipulation and simulation* of the software performing legally relevant functions.
- *Harmonise software examination and documentation* by the Notified Bodies as part of the type approval and testing procedures for NAWIs and related modules or peripheral devices.

This approach takes into consideration the interests and responsibilities of both, the manufacturer and the Notified Body. The manufacturer has a vital interest not only in the flexibility of his instruments but also in its protection against any misuse as far as he is liable for his product; this includes his responsibility for the conformity of the individual instrument to the approved type. The Notified Body by law is forced to examine thoroughly the conformity of a type with EC regulations and the measures taken to protect the customer of an instrument against wrong measurement, unintentional misuse and fraudulent use.

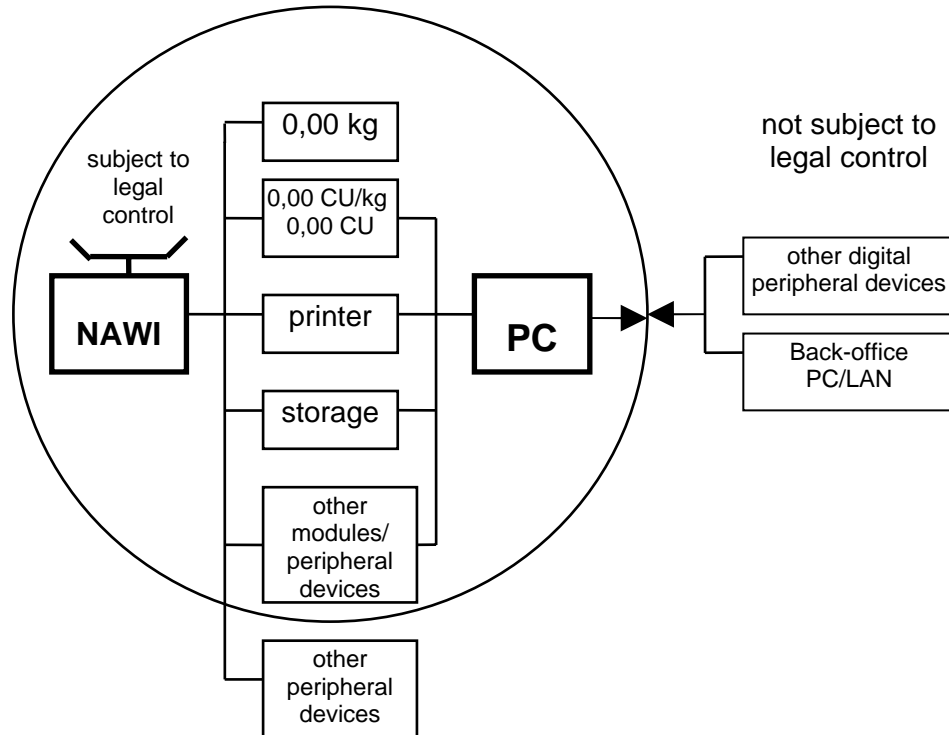
## 1.3 Scope

This guideline specifies *basic requirements* to be applied to software for freeprogrammable, PC-based modules or peripheral devices which are linked to, or from part of, NAWIs subject to legal control.

The Figures 1 and 2 schematically illustrate the structures of the hardware and software of a PC-based weighing system comprising devices and functions subject to legal control (inside the circles) and others not subject to legal control (outside the circles). Both figures are intended to serve as examples in order to demonstrate the basic principles of this guideline rather than as sophisticated models that cover all possible technical solutions. Therefore, they have to be

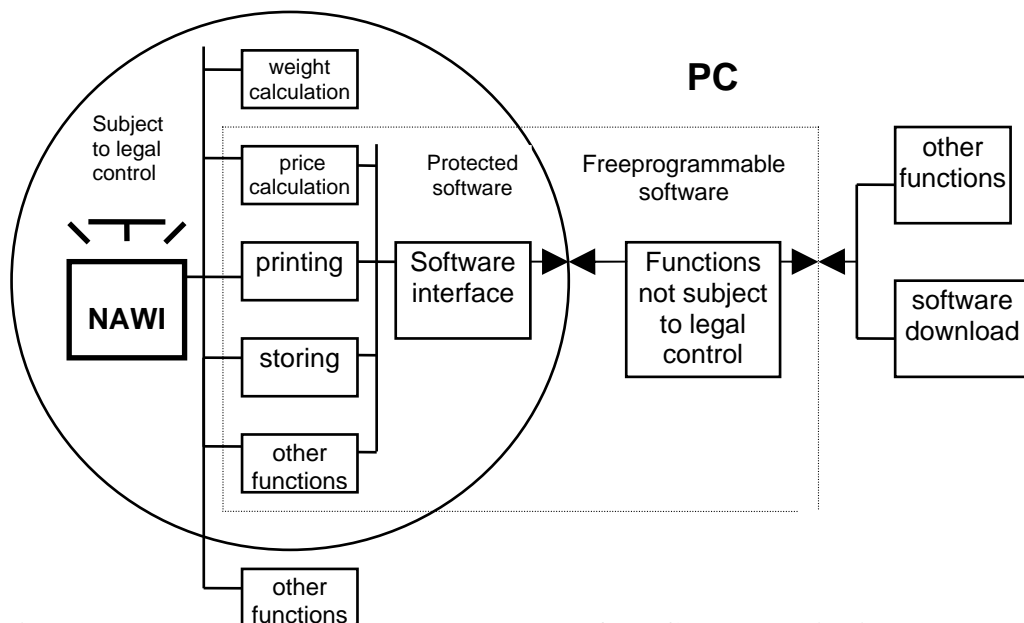
interpreted with close regard to the requirements of the Directive 90/384/EEC and the European Standard EN 45501, respectively.

HARDWARE



**Figure 1: Example of a hardware structure of a PC-based weighing system; CU current unit**

SOFTWARE



**Figure 2: Example of a software structure of a PC-based weighing system**

The basic instrument is the NAWI comprising at least a load cell, a load receptor, a microprocessor system including an A/D converter and a weight display. The basic instrument, if need be in combination with separate modules or peripheral devices, may perform further functions subject to legal control, such as price calculation, price indication, printing or storing of weighing results and other functions, e.g. tare balancing, preset tare. Other peripheral devices not subject to legal control may be connected to the NAWI if the respective hardware interfaces are protective in the sense of EN 45501, No 5.3.6.1.

A freeprogrammable PC-based module or peripheral device will *not* be subject to legal control if it communicates with the NAWI via a protective *hardware interface* and if it does not carry out *any* legally relevant function.

But today, freeprogrammable modules and peripheral devices - for example POS devices - take over more and more **legally relevant functions from the NAWI, eg printing of weighing results or price calculation**. In this case, the *hardware* of the PC-based device is subject to legal control (see Figure 1). The *software* of such a device performs both functions subject to legal control and others not subject to legal control (see Figure 2).

The scope of this guide is to define the basic requirements to be met by the software of a PC-based device in order to have a *freeprogrammable* software part which can be adapted to the special needs of a customer and a *protected and approved* software part realising legally relevant functions which can only be changed with the knowledge and consent of the responsible Notified Body. This can be achieved by realising a *software interface* between the protected software part and the freeprogrammable part (see Figure 2 and No 3.2) which itself is protective in the sense of EN 45501, No 5.3.6.1. The software download for the PC-based device may then even be performed from a back-office PC within a local area network (LAN); also any other digital peripheral device not subject to legal control may be connected to the PC via an arbitrary (not necessarily protective) hardware interface.

Of course, the manufacturer of the instrument is free to declare the entire software to be subject to legal control. In this case the complete PC-based device including the software will be subject to legal control (ie hardware and software of the device are completely inside the circle, see Figures 1 and 2) and *any* change of the software would have to be announced to the responsible Notified Body according to Annex II, No 1.7, Directive 90/384/EEC (see Note 2 under No 3.2). In this case, of course, all hardware interfaces of the PC-based device to other digital peripheral devices would have to be protective in the sense of EN 45501, No 5.3.6.1.

## 1.4 Conception

The conception of this guideline is the following:

- Definition of the most important terms in section 2 '*Terminology*'.
- Formulation of four essential requirements for the software of freeprogrammable modules or devices connected to NAWIs subject to legal control in section 3 '*Software requirements*'.
- *Notes* to the essential requirements to support their uniform interpretation.
- Suggestion of *acceptable solutions* to the manufacturer to demonstrate how he can meet the essential requirements. The manufacturer is free to choose different solutions if he can prove that with his solutions the essential requirements are met as well.

- Proposals to the Notified Bodies concerning a *report* on the software examination, see section 4, and the *specifications* required in the type approval certificate (TAC) of the complete instrument or in the test certificate (TC) of the freeprogrammable module or peripheral device, see section 5.

The acceptance of certificates issued by other Notified Bodies is greatly enhanced if the results of the software examination are documented properly.

- This guideline is intended to serve as a *preliminary document* stating *basic software requirements* for a special type of measuring instrument, see section 6. On the one hand, it surely needs to be revised after some time (e.g. after one year), when enough experience and knowledge have been gained by the responsible Notified Bodies; on the other hand it shall not anticipate general software requirements for all classes of measuring instruments which will be worked out by WELMEC WG7 or other WELMEC working groups.

## 2 Terminology

### 2.1 Legally relevant software

Program parts and data that form, by declaration of the manufacturer and by approval of the notified body, the software subject to legal control, see Figures 2 and 3.

#### Legally relevant program parts

Parts of the legally relevant software which realise functions subject to legal control, see Figure 3.

#### Legally relevant data

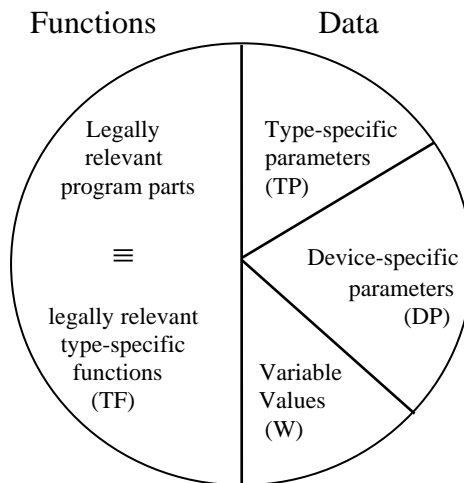
Parameters and data subject to legal control; according to Figure 3, the following types of legally relevant data can be distinguished:

- *Type-specific parameters* which depend on the special type of instrument only. Type-specific parameters are fixed at the type approval of the instrument.
- *Device-specific parameters* which depend on the individual device or instruments; device-specific parameters comprise *calibration parameters* (e.g. of span adjustment, other adjustments or corrections) and *configuration parameters* (e.g. Max, Min, e, d, units of measurement). Device-specific parameters are adjustable or selectable only in a special operational mode of the instrument. Some device-specific parameters may also be type-specific.

*Variable values* which depend on the measurement (weighing) process itself. Variable values comprise *processed variable values* which are still under process of the legally relevant program parts and *final variable values* which are the final results of the measurement (weighing) process.

Examples of legally relevant functions and data are given in Table 1.

Legally relevant software



**Figure 3:** Schematic representation of the legally relevant software comprising legally relevant program parts (functions) and data

**Table 1:** Examples of legally relevant functions and data

<b>Functions/data</b>	<b>Type</b>
weight calculation	TF, TP
stability analysis	TF, TP
price calculation	TF, TP
rounding algorithm for price	TF, TP
span (sensitivity)	DP
corrections for non-linearity	DP (TP)
Max, Min, e, d	DP (TP)
units of measurement (eg g, kg)	DP (TP)
weight value as displayed (rounded to multiples of e)	W
tare, preset tare	W
unit price, price to pay	W
weight value in internal resolution	W
status signals (eg zero indication, stability of equilibrium)	W

## 2.2 Changes of software

### Unintentional changes

Changes of program parts or data subject to legal control that are unintentionally performed *by the user* of the instrument.

### Intentional changes with common software tools

Changes of the legally relevant software that are performed using functions of its own or software tools and know-how commonly available to the general public.

### Intentional changes with special software tools

Manipulation or simulation of the legally relevant software that is performed using software tools and know-how not commonly available to the general public.

## 2.3 Protection of software

### Protected software

Software, including type- and device-specific parameters, a change of which either is not possible or is detected and made evident e.g. by an audit trail.

### Audit trail (“Footprint”)

An electronic count and/or information record of the changes to the device-specific parameters. An audit trail can be realised e.g. as an ‘Event counter’ or as an ‘Event logger’.

### Event counter

A non-resettable counter that increments once each time a special operational mode of the instrument is entered and one or more changes are made to device-specific parameters.

### Event logger

A file containing a series of records where each record contains the number from the event counter corresponding to the change to a device-specific parameter, the identification of the parameter that was changed, the time and date when the parameter was changed and the new value of the parameter.

## 3 Software requirements

### 3.1 Protection of the legally relevant software

<p><i>The legally relevant software shall be protected against intentional changes with common software tools</i></p>
---

#### Note 1

The program parts and data of the legally relevant software will be regarded as sufficiently protected against *unintentional changes* if the above requirement is met.

#### Note 2

The protection against *intentional changes with special software tools* is not the object of these requirements, as those changes are considered as criminal actions which are covered by existing laws.

Note 3

It can normally be assumed that it is not possible to influence legally relevant data - especially processed variable values - as long as they are processed by a program which fulfils the requirements under No 3.1 and No 3.2. However, if legally relevant data - especially final variable values - will be transmitted out of the protected software part for applications or functions subject to legal control, see Figure 2, they shall be secured in order to meet EN 45501, No 5.3.6.3.

Note 4

The part of the legally relevant software exclusively dealing with final variable values will be regarded as sufficiently protected, if these variable values cannot be changed with common software tools.

Note 5

At the moment, for example, all kinds of text editors are regarded as common software tools.

Example of an acceptable solution

<b>Objective</b>	<b>Acceptable solution for legally relevant software</b>	<b>Documentation for type approval</b>	<b>Examples for checking at type approval</b>
<p>Protection of the relevant functions of the measuring instrument while measurement is active.</p> <p>Protection against circumventing of software interfaces (see section 3.2).</p>	<ul style="list-style-type: none"> <li>◦ closed shell of the programs subject to legal control, only controlled access to the operating system for the user</li> <li>◦ communication between programs subject to legal control and others via software interfaces according section 3.2</li> </ul>	<ul style="list-style-type: none"> <li>◦ complete set of commands (from keyboard or any interface) and the meaning of each command</li> <li>◦ declaration of the completeness of the documented command set</li> </ul>	<ul style="list-style-type: none"> <li>◦ practical test of the shell by checking, whether all commands operate as documented</li> <li>◦ check, whether the declaration for completeness is given</li> <li>◦ verify the protection means by using a text editor</li> </ul>
<p>Protection against intended changes of the legally relevant program and the type-specific and device-specific parameters</p>	<ul style="list-style-type: none"> <li>◦ checksum and audit trail over machine code of legally relevant program parts and type-specific parameters</li> <li>◦ checksum and audit trail over device-specific parameters</li> <li>◦ no start if code is falsified</li> </ul>	<ul style="list-style-type: none"> <li>◦ declaration that checksum(s) are generated</li> <li>◦ documentation of the manufacturers selected solution</li> </ul>	<ul style="list-style-type: none"> <li>◦ check, whether checksum(s) are generated and comply with the documentation</li> <li>◦ verify the protection means by using a text editor</li> </ul>

### 3.2 Software interfaces

*Interfaces between the legally relevant software and the software parts not subject to legal control shall be protective*

#### Note 1

If parts of software exist besides the legally relevant parts, these parts shall be separated in a sense that they communicate via a software interface, see Figure 2. A software interface is defined as being protective.

- if in accordance with EN 45501, No 5.3.6.1, only a defined set of parameters and functions of the legally relevant software part can be influenced via this interface and
- if both parts do not exchange information via any other link.

Software interfaces are part of the legally relevant software. They comprise program modules and data structures.

#### Note 2

Software interfaces need not be protective if the manufacturer will announce *any* change of the software (including the legally irrelevant part) of an EC type-approved instrument to the notified body according to EC directive 90/384, Annex II, No 1.7. In this case, the software identification, cf. No 3.3, must comprise the entire program.

#### Note 3

Circumventing the protective interface *by the user* is considered as a criminal action if the software is protected in the sense of No 3.1.

#### Examples of an acceptable solution

<b>Acceptable solution for legally relevant software</b>	<b>Documentation for type approval</b>	<b>Examples for checking at type approval</b>
<ul style="list-style-type: none"> <li>◦ definition of program modules used to handle legally relevant functions and data</li> <li>◦ definition of functions which may be released via the protective interface</li> <li>◦ definition of data which may be exchanged via the protective interface</li> </ul>	<ul style="list-style-type: none"> <li>◦ short functional description of the legally relevant program modules</li> <li>◦ complete list of the legally relevant functions and data</li> <li>◦ declaration of completeness of these lists</li> </ul>	<ul style="list-style-type: none"> <li>◦ check, whether the functional description is conclusive</li> <li>◦ check, whether all documented functions or data released or exchanged via the protective interface are allowed</li> <li>◦ check, whether the declaration for completeness is given</li> </ul>

### 3.3 Software identification

*There must be a software identification, comprising the legally relevant program parts and parameters, which is capable of being confirmed at verification.*

#### Note 1

The software identification may be split into two parts, one comprising the non-adjustable, type-specific functions and parameters, the other one comprising the device-specific parameters, see Figure 3.

#### Note 2

The operating system of the PC and auxiliary software, such as video drivers, printer drivers or hard disk drivers, need not be included in the software identification. However, special application software made by or by order of the manufacturer of the instrument shall be included in the software identification if those program parts affect the printer or display subject to legal control (e.g. software parts realising the layout and printing of a receipt, see Figure 2).

#### Example of an acceptable solution

<b>Acceptable solution for legally relevant software</b>	<b>Documentation for type approval</b>	<b>Examples for checking at type approval</b>
<ul style="list-style-type: none"> <li>◦ checksum (or other signature) over machine code which represents the legally relevant program parts and type-specific parameters</li> </ul>	<ul style="list-style-type: none"> <li>◦ documentation of the manufacturers selected solution</li> </ul>	<ul style="list-style-type: none"> <li>◦ check, whether the checksum(s) or other signature(s) are generated and may be confirmed at verification</li> </ul>
<ul style="list-style-type: none"> <li>◦ checksum (or other signature) over device-specific parameters</li> </ul>	<ul style="list-style-type: none"> <li>◦ documentation of the manufacturers selected solution</li> </ul>	<ul style="list-style-type: none"> <li>◦ check, whether the checksum(s) or other signature(s) are generated and may be confirmed at verification, eg by an audit trail</li> </ul>

### 3.4 Documentation

*The documentation shall describe:*

- All legally relevant parts and parameters of the software.
- The functions of these parts.
- The complete set of commands to be exchanged via the protective software interface.
- A written declaration of completeness of the list of the legally relevant functions and parameters and the documented set of commands.
- The securing measures (e.g. checksum, software identification, audit trail).
- The instructions in order to check the legally relevant software at verification.
- A written declaration that the standard EN 45501:1992/AC 1993 has been adopted.

#### **4 Report on the software examination**

The software examination by the Notified Body is to be documented in a *short* report which can be made available to other Notified Bodies on their request.

The report shall contain:

- A reference to the type of PC-based, freeprogrammable instrument, module or peripheral device used for the examination of the software. If a certificate (TAC or TC) was issued for that device, the respective certificate number should also be referred to.
- A list of the documents concerning the software supplied by the manufacturer and examined by the Notified Body (including date and identification No).
- A list of programs and program modules, including their identification numbers, which form the legally relevant software.
- A checklist containing the examinations performed in order to verify that the requirements under No 3.1 to 3.4 are met. The checklist shall comprise all checks mentioned under 'Examples for checking at type approval' in the tables under No 3.1 to 3.3 and all points mentioned under No 3.4. If the manufacturer offers a solution differing from the given 'Examples of an acceptable solution', the reasons for accepting this solution shall be given.

#### **5 Required specifications in certificates**

The type-approval certificate (TAC) of the complete, freeprogrammable NAWI or the test certificate (TC) of the freeprogrammable module or peripheral device of a NAWI shall contain the following specifications:

- A statement that there exist two separate software parts, one part representing the legally relevant software and the other one realising functions not subject to legal control.
- A statement that the legally relevant software meets the requirements No 3.1 to 3.4 of the 'WELMEC Guide for examining software of non-automatic weighing instruments (NAWI)'
- A *short* functional description of the legally relevant software, including e.g. keyboard interfaces, terminal interfaces, hard disk interfaces and the software interface (mentioning the different interfaces and their functions is sufficient).
- The identification number(s) of the legally relevant software
- A list of a summary of the software documents of the manufacturer (reference to the report on the software examination, cf. section 4)
- Information for verification:
  - How to verify the software identification
  - How to get access to detected software changes made evident e.g. by an audit trail

#### **6 Preliminary procedure**

These requirements are valid until general software requirements for measuring instruments under legal control will be passed by WELMEC. For the time being, only the functional description of the software is examined according to the requirements under No 3.1 to No 3.4.