Appendix D

National Type Evaluation Technical Committee Software Sector

October 17 - 18, 2007 – Little Rock, Arkansas Meeting Summary

Agenda Items

Meeting Minutes

Jim Truex called the meeting to order at 8:00 on October 17, 2007. All registered participants attended. Jim explained that the Sector attempts to build consensus and then explained the voting procedures, if needed. He asked everyone to introduce himself or herself.

Carryover Items

1.a. NTETC Software Sector Mission

Source: NCWM Board of Directors

Background: In 2005 the Board of Directors established a National Type Evaluation Technical Committee (NTETC) Software Sector. A mission statement for the Sector was developed at that time.

Mission of the Software Sector:

- Develop a clear understanding of the use of software in today's weighing and measuring instruments.
- Develop NIST Handbook 44 specifications and requirements, as needed, for software incorporated into weighing and measuring devices. This may include tools for field verification, security requirements, identification, etc.
- Develop NCWM Publication 14 checklist criteria, as needed, for the evaluation of software incorporated into weighing and measuring devices, including marking, security, metrologically significant functions, etc.
- Assist in the development of training guidelines for W&M officials in verifying software as compliant to applicable requirements and traceable to a NTEP Certificate. Training aids to educate manufacturers, designers, service technicians and end users may also be considered.

From previous meeting:

Discussion: The Chair asked the question: Is the Sector comfortable with the Mission Statement?

The Sector discussed the process of other NTETC sectors, the NCWM structure and how/why, the Software Sector was developed. After some lengthy discussion by the Sector, there was consensus among the Sector members that the Mission Statement is correct. However, the Sector noted that there is a very broad range of items listed in the Statement. The Sector agreed that the steps in the Mission Statement are correct. The steps appear to build on each other in an orderly progression. It was further agreed that whenever possible items will be addressed in the sequence of the Mission Statement.

The Chair noted that the scope of this Sector is somewhat broader than some other sectors. The work of this Sector is more closely aligned to that of the Grain Analyzer Sector in that focus is on development of possible language for:

- NIST Handbook 44,
- checklist criteria for NCWM Publication 14, and
- appropriate field guidelines.

Comments from October meeting:
Jim Truex noted there would be an attempt to follow the four bullet items above in order from the top down when discussing agenda items. Focus should begin with any possible impact on NIST Handbook 44.

1.b. NCWM/NTEP Policies – Issuing CCs for Software

Source: NCWM Reports

Background: Excerpts of reports from the 1995-1998 Executive Committee were provided to NTETC Software Sector members at their April 2006 meeting. The chair asked the Sector to review the following NTEP policy decision adopted by the NCWM in 1998 relative to the issuance of a separate Certificate of Conformance (CC) for software.

> The NCWM has struggled with software issues for many years. Prior to 1995, NTEP had evaluated stand alone software (e.g., weigh-in/weigh-out, POS, and batch controller software) and, in some cases, had issued CCs for stand alone software. The Board established a software work group to study the issues and make recommendations.
>
> Many issues were discussed by the work group, including: first indication of the final quantity, metrologically significant software, definitions, software marking, software checklist evaluation, a software EPO for the field inspector, user programmable software, and third party software. According to conference reports, it seems in 1997 some concerns were raised about the direction of the work group. In 1997, after the Annual Meeting, a new Software Work Group was appointed by the NCWM chair.
>
> **During the 1998 NCWM, the following recommendation was adopted as NTEP policy:**
>
> - **"Software, regardless of its form, shall not be subject to evaluation for the purpose of receiving a separate, software Certificate of Conformance from the National Type Evaluation Program."**
> - **"Remove all of the software categories from the index of NCWM Publication 5, NTEP Index of Device Evaluations."**
> - **"Reclassify all existing software CCs according to their applicable device categories."**

The policy is still in effect today.

Also noteworthy is a statement in Section C of NCWM Publication 14, Administrative Policy. It states: "In general, type evaluations will be conducted on all equipment that affect the measurement process or the validity of the transaction (e.g., electronic cash registers interfaced with scales and service station consoles interfaced with retail fuel dispensers); and all equipment to the point of the first indicated or recorded representation of the final quantity on which the transaction will be based."

Discussion: At this point in time, NTEP evaluates a "software-based device" as a functional device. The performance of the device is evaluated.

There was a suggestion from the floor that the 1998 policy be amended. If this is done, then the Sector can move toward the other steps in the process.

Discussion from the floor is on how to or if there needs to be a change to the device type in the FOR box.

The consensus of the Sector is that the current NCWM/NTEP policy should be changed.

From previous meeting:

Software Requiring a Separate CC: Software which is implemented as an add-on to other NTEP Certified main elements to create a weighing or measuring system and its metrological functions are significant in determining the first indication of the final quantity. Such software is considered to be a main element of the system requiring a separate CC. (traceability to an NTEP CC)

NOTE: OEM software ***may*** be added to an existing CC or have a stand-alone CC with applicable applications (e.g., a manufacturer adding a software upgrade to their ECR or point-of-sale system, vehicle scale weigh-in/weigh-out software added as a feature to an indicating element, automatic bulk weighing, liquid-measuring device loading racks, etc.) and minimum system requirements for "type P" devices (see proposed software definition below). It may be possible for a manufacturer to submit a single application for both hardware and software contained in the same device. A single CC would be issued.

In this instance, OEM refers to a 3rd party. The request to add software could be made by the original CC holder on behalf of the 3rd party. Alternatively, a new CC could be created that refers to the original CC and simply lists the new portions that were examined.

The Sector recommendation will be submitted to the NTEP Committee.
This item has not yet been submitted to the NTEP Committee for review. It is planned for this to happen during the NCWM Interim Meeting in January 2008.

October Meeting Discussion:
Some concerns were raised by the California laboratory regarding this recommendation. During the course of the discussion, these concerns were addressed and resolved.

Don Onwiler indicated that this may be a technical policy that needs to be inserted into each different volume or chapter of NCWM Publication 14 or it may need to be placed in the Administrative Policy volume.

It was agreed that overall, there would be no change to what is currently being done by NTEP and the labs to certify devices, however; the device type or name of the device certified would be changed.

Recommendation from the Sector to the NTEP Committee:

The Sector recommended the following language to be submitted to the NTEP Committee as a policy change.

Software Requiring a Separate CC: Software, which is implemented as an add-on to other NTEP Certified main elements to create a weighing or measuring system and its metrological functions, are significant in determining the first indication of the final quantity. Such software is considered a main element of the system requiring traceability to an NTEP CC.

NOTE: OEM software ***may*** be added to an existing CC or have a stand-alone CC with applicable applications (e.g., a manufacturer adding a software upgrade to their ECR or point-of-sale system, vehicle scale weigh-in/weigh-out software added as a feature to an indicating element, automatic bulk weighing, liquid-measuring device loading racks, etc.) and minimum system requirements for "type P" devices (see proposed software definition below). It may be possible for a manufacturer to submit a single application for both hardware and software contained in the same device. A single CC would be issued.

In this instance, OEM refers to a 3rd party. The request to add software could be made by the original CC holder on behalf of the 3rd party. Alternatively, a new CC could be created that refers to the original CC and simply lists the new portions that were examined.

2. Definitions for Software-Based Devices

Source: NTETC Software Sector

Background: Discussed was marking and G-S.1.1. It was initially suggested that "not built-for-purpose" be removed from the wording in NIST HB 44 G-S.1.1. However, after further discussion this may not be the correct or final decision. There is no definition for a not built-for-purpose device in HB 44. The current HB 44 definition for a built-for-purpose device reads:

> **Built-for-purpose device**: Any main device or element which was manufactured with the intent that it be used as, or part of, a weighing or measuring device or system. [1.10] (Added 2003)

There was also the suggestion to use the definitions from the WELMEC document for Type P and Type U instruments. They were modified by the Sector. It was also suggested that a list of examples be provided.

Draft definitions for consideration:

Built-for-purpose weighing or measuring instrument (device) (type P): A weighing or *measuring Instrument (device)* designed and built specially for the task in-hand. Accordingly the embedded software is assumed to be designed for the specific task. It may contain many components also used in PCs, e.g., motherboard, memory card, etc.

A weighing or measuring instrument (device) using a universal Computer (type U): *A weighing or measuring Instrument (device)* that uses a general-purpose computer, usually a PC-based system, for performing metrologically significant functions.

Examples:
Type U
Weigh-in, Weigh-out
Open Architecture

Discussion: The Sector agrees that the NTEP CC should reflect "software" is a separate main element. If this is true then there needs to be definition.

The Sector agrees that this change in policy and appearance on CC's does not have a major impact on our current type evaluation process.

MC cites three main areas of software: sensing physical phenomena (mass or volume), computational, controlling the system.

After a lengthy discussion related to the terms "built-for-purpose" and "not-built-for-purpose", the Sector agreed that these terms were not clear and should be replaced with the terminology proposed below.

A main reference point that the Sector used in this discussion was OIML R 76 *Non-automatic weighing instruments* sub-sections 5.5.1. (Type P) and 5.5.2. (Type U).

New Definition:

Electronic devices, software-based. Weighing and measuring devices or systems that use metrological software to facilitate compliance with Handbook 44. This includes:

(a) **Embedded software devices (Type P).** A device or element with software used in a fixed hardware and software environment that cannot be modified or uploaded via any interface without breaking a security seal or other approved means for providing security, and will be called a "P", or

(b) **Programmable or loadable metrological software devices (Type U).** A personal computer or other device and/or element with PC components with programmable or loadable metrological software, and will be called "U". A "U" is assumed if the conditions for embedded software devices are not met.

October Meeting Discussion:
After some discussion on this item the Sector agreed to forward the recommendation to the S&T Committee.

Recommendation from the Sector to the S&T Committee:

The Sector recommended that the following definitions be submitted to the S&T Committee as a developing item and be considered for inclusion in NIST Handbook 44.

Electronic devices, software-based. Weighing and measuring devices or systems that use metrological software to facilitate compliance with Handbook 44. This includes:

(c) **Embedded software devices (Type P) aka built for purpose.** A device or element with software used in a fixed hardware and software environment that cannot be modified or uploaded via any interface without breaking a security seal or other approved means for providing security, and will be called a "P", or

(d) **Programmable or loadable metrological software devices (Type U) aka not built for purpose.** A personal computer or other device and/or element with PC components with programmable or loadable metrological software, and will be called "U". A "U" is assumed if the conditions for embedded software devices are not met.

3. Software Identification/Markings

Source: NTETC Software Sector

Background: At the last meeting there was discussion on specific sections of the WELMEC document that deal with TYPE P and TYPE U requirements. The comments and recommendations under consideration are contained in the following.

Discussion: There was lengthy discussion on the value and merits of markings. This included the possible differences in some types of devices and marking requirements. After hearing several proposals the Sector agreed to the following recommendation.

Technical changes represented below:

1. CC No. must be continuously displayed or marked,
2. Version must be software generated, not hard marked,
3. Version required for embedded (Type P),
4. Print option created,
5. Command or operator action option created,
6. Type P must display or hard mark make, model, S.N.

From Previous Meeting:
The Sector will forward these items, when completed, to the Regional S&T committees for consideration.

October Meeting Comments:
This section needs to be completed with the actual changes to HB 44 sections. There is some concern with the note that is contained below Type P device.

There may be the need to have a delineation of devices with "firmware." An exception may need to be made for a device that is "integral and blind." It is possible that NTEP needs to determine if the "software" is integral and does not need to be identified. Need to know the rules up front.

Metrologically significant software shall be clearly identified with the software version. The identification may consist of more than one part but one part shall be only dedicated for the metrologically significant portion.

Measurement Canada commented on "primary sensing elements" and exemption from certain requirements (digital load cells and devices with correction methods). This is needed to prevent a "black box" which could be added in between other main elements and then be exempt from certain requirements.

Difference may be that the Digital Load Cell has been evaluated integral, while the digital J-Box can be modified or built with various components and characterized in the field.

One manufacturer still has a problem with the exemption, (footnote 3 below) and as an example used a smart J-box.

The "Via Menu (display) or Print option" may be supplemental for devices that use the hard-marked or continuously displayed identification method for the NTEP CC Make/Model, Serial No. information.

Metrologically Significant software shall be clearly identified with the software version. The identification may consist of more than one part but one part shall be only dedicated for the metrologically significant portion.

Currently there is no specification for permanence of the marking for software (the CC No. on the screen). This will need to be addressed by the Sector.

Developing Recommendation from the Sector to the S&T Committee:

The Sector recommended that the following marking information be submitted to the S&T Committee as a developing item and be considered for inclusion in NIST Handbook 44.

TYPE P shall meet at least one of the methods in each column:

Method	NTEP CC No.	Make/Model/Serial No.	Software Version/Revision[3]
Hard-Marked	X	X	Not Acceptable
Continuously Displayed	X	X	X
By Command or Operator Action	Not Acceptable	Not Acceptable	X[4]

[3] If the manufacture declares that the primary sensing element "software" is integral, has no end user interface and no print capability, the element may be considered exempt from the marking requirement for version/revision.

Example: primary sensing element may be P.D. meter with correction, digital load cell. (only for reference, not limiting)

[4] Information on how to obtain the Version/Revision shall be included on the NTEP CC.

TYPE U shall meet at least one of the methods in each column:

Method	NTEP CC No.	Make/Model	Software Version/Revision
Hard-Marked	X[1]	X	Not Acceptable
Continuously Displayed	X	X	X
Via Menu (Display) or Print Option	Not Acceptable	X[2]	X[2]

[1] Only if no means of displaying this information is available.

[2] Information on how to obtain Make/Model, Version/Revision shall be included on the NTEP CC.

4. Identification of Certified Software

Source: NTETC Software Sector

Previous meeting notes:

Separation of software
Separation of metrological and application software as described in the OIML documents is maintained.

5.2.1.2. Separation of software parts

Requirement (a): All software modules (programs, subroutines, objects etc.) that perform metrologically significant functions or that contain metrologically significant data domains form the metrologically significant software part of a measuring instrument (device or sub-assembly). The conformity requirement applies to this part (see 5.2.5.) and it shall be made identifiable as described in 5.1.1.

If the separation of the software is not possible or needed, the software is metrologically significant as a whole. Segregation of parameters is currently allowed. (see table of sealable parameters)

October Meeting Discussion:
The sector agreed that the title of this item needs to be changed to "Identification of Certified Software. Currently, used are version no., ID no., and serial no. However; there is no physical tie to the actual software. Some international documents, like the WELMEC document tell how to tie the ID to the software. These include:

Possible methods: (not limited to)
- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption

The question remains is there some method to give the W&M inspector information that something has changed? How can the W&M inspector easily identify an NTEP Certified version?

Required Documentation:
The documentation shall list the software identifications and describe how the software identification is created, how it is inextricably linked to the software itself, how it may be accessed for viewing and how it is structured in order to differentiate between version changes with and without requiring a type approval**.**

NTEP strongly recommends that metrological software be separated from non-metrological software for ease of identification and evaluation.

Separation of software parts
All software modules (programs, subroutines, objects etc.) that perform metrologically significant functions or that contain metrologically significant data domains form the metrologically significant software part of a measuring instrument (device or sub-assembly). The conformity requirement applies to all parts and parts shall be marked according to Section G-S.X.X.

If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.

Segregation of parameters is currently allowed. (see table of sealable parameters)

Conclusion from the October Meeting: The Sector will continue to develop this item.

5. Software Protection/Security

The Sector spent a significant amount of time reviewing and revamping previous work. OIML and WELMEC documents were researched. The following are draft Publication 14 checklist criteria for consideration at the next meeting.

Building Publication 14 Checklist information:

(Reference Information taken from OIML R 76-2 Draft Document)

Section YY: Additional requirements for software-controlled electronic devices

YY.1. Devices with embedded software: Type P (Built for purpose)

For instruments and modules with embedded software, the manufacturer shall describe or declare that the software of the instrument or module is embedded, i.e., it is used in a fixed hardware and software environment and cannot be modified or uploaded via any interface or by other means after securing and/or verification.

In addition to all other required documentation the manufacturer shall submit the following documentation:

- description of the metrologically significant functions,
- software identification that is clearly assigned to the metrologically significant functions, and
- securing measures foreseen to provide for evidence of an intervention.

The software identification shall be provided by the instrument and listed in the NTEP Certificate of Conformance.

Acceptable solution:

The software identification is provided by either:

- in the normal operation mode a clearly identified operation of a physical or soft key, button, or switch, or
- in the normal operation mode a continuously displayed version number or checksum, etc., accompanied in both cases by clear instructions how to check the actual software identification against the reference number (as listed in the NTEP CC) marked on or displayed by the instrument.

YY.2. Personal computers, instruments with PC components, and other instruments, devices, modules, and elements with programmable or loadable metrologically significant software: Type U (not built for purpose)

Personal computers and other instruments/devices with programmable or loadable software may be used as indicators, terminals, data storage devices, peripheral devices, etc. if the following additional requirements are met.

Note: *Although these devices may be complete weighing instruments with loadable software or PC-based modules and components, etc. they will in the following simply be called "PC". A "PC" is always assumed if the conditions for embedded software are not fulfilled.*

YY.2.1. Hardware requirements

PCs as modules incorporating the metrologically relevant analogue component(s) shall be treated according to Table ZZ, categories 1 and 2.

PCs acting as a purely digital module without incorporating metrologically relevant analogue components (e.g., used as terminals or price-computing point-of-sale devices) shall be treated according to Table ZZ, categories 3 and 4.

PCs used as purely digital peripheral devices shall be treated according to Table ZZ, category 5.

Table ZZ also specifies how detailed the documentation to be submitted for both analogue and digital components of the PC shall be depending on the respective category (description of power supply, type of interfaces, motherboard, housing, etc.).

Table ZZ: Tests and required documentation for PCs used as modules or peripheral devices

Table ZZ. Tests and Required Documentation for PCs Used as Modules or Peripheral Devices

Category		Necessary Tests	Documentation	Remarks
No.	**Description**		**Hardware Components**	
1	PC as a module, primary indications on the monitor, PC incorporates the metrologically relevant analogue components (ADC) on a slot mounted circuit print board that is not shielded ("open device"), power supply device for the ADC from the PC or PC-bus system	ADC and PC tested as unit: tests as for indicators according to Annex C; the pattern shall be equipped with the maximum possible configuration (maximum power consumption)	ADC: detailed as for instruments and modules (circuit diagrams, layouts, descriptions etc.) PC: detailed as for instruments and modules (manufacturer, type of PC, type of housing, types of all modules, electronic devices and components including power supply device, data sheets, manuals, etc.)	Influences on the ADC from the PC possible (temperature, electromagnetic interference (EMC))
2	PC as a module, primary indications on the monitor, PC incorporates the ADC, but the built-in ADC has a shielded housing ("closed device"), power supply device for the ADC from the PC, but not via the PC-bus system	ADC and PC as unit: tests as for indicators according to Annex C; the pattern shall be equipped with the maximum possible configuration (maximum power consumption)	ADC: detailed as for instruments and modules (circuit diagrams, layouts, descriptions etc.) PC: Power supply device: detailed as for instruments and modules (manufacturer, type, data sheet) Other parts: only general description or information necessary concerning the form of housing, motherboard, processor type, RAM, floppy and hard disk drives, controller boards, video controller, interfaces, monitor, keyboard, etc.	Influences on the ADC from the power supply device of the PC possible (temperature, EMC), other influences from the PC not critical, new EMC tests (PC) necessary if the power supply device is changed
3	PC as purely digital module, primary indications on the monitor, ADC outside the PC in a separate housing, power supply device for the ADC from the PC	ADC: tests as for indicators according to Annex C using the monitor of the PC for the primary indications PC: according to 3.10.2	ADC: as for category 2 PC: Power supply device as for category 2, other parts as for category 4	Influence (only EMC) on the ADC from the power supply device of the PC possible Other influences from the PC not possible or not critical New EMC tests (PC) necessary if the power supply device is changed
4	PC as purely digital module, primary indication on the monitor, ADC outside the PC in a separate housing having its own power supply device	ADC: as for category 3 PC: as for category 3	ADC: as for category 2 PC: Only general description or information necessary, e.g., concerning type of motherboard, processor type, RAM, floppy and hard disk drives, controller boards, video controller, interfaces, monitor, keyboard	Influences (temperature, EMC) on the ADC from the PC not possible
5	PC as purely digital peripheral device	PC: according to 3.10.3	PC: as for category 4	

Meaning of the abbreviations used in Table ZZ: PC – Personal Computer, ADC – Relevant analogue component(s), including Analogue/Digital-Converter (see Figure 1), EMC – Electromagnetic Compatibility.

YY.2.2. Software requirements
The metrologically significant software of a PC, i.e., the software that is critical for measurement characteristics, measurement data and metrologically important parameters stored or transmitted, is considered as an essential part of a weighing instrument and shall be examined according to Annex G.2. The metrologically significant software shall meet the following requirements.

a. The metrologically significant software shall be adequately protected against accidental or intentional changes. Evidence of an intervention such as changing, uploading or circumventing the metrologically significant software shall be available until the next verification or comparable official inspection. This requirement implies that:

> The protection against intentional changes with special software tools is not the object of these requirements, because this is considered as criminal action. It can normally be assumed that it is not possible to influence metrologically significant parameters and data – especially processed variable values – as long as they are processed by a program which fulfils these requirements. However, if metrologically significant parameters and data – especially final variable values – will be transmitted out of the protected software part for applications or functions subject to legal control, they shall be secured to meet the requirements of 5.3.6.3.

The metrologically significant software with all data, parameters, variable values, etc., will be regarded as sufficiently protected, if they cannot be changed with common software tools. At the moment, for example, all kinds of text editors are regarded as common software tools.

Acceptable solution:

> After program start automatic calculation of a checksum over the machine code of the complete metrologically significant software (at least a CRC-16 checksum with hidden polynomial) and comparison of the result with a stored fixed value. No start if the machine code is falsified.

b. When there is associated software which provides other functions besides the measuring function(s), the metrologically significant software shall be identifiable and shall not be inadmissibly influenced by the associated software.

This requirement implies that:

> Associated software is separated from the metrologically significant software in a sense that they communicate via a software interface.

A software interface is regarded as being protective if:

- in accordance with 5.3.6.1 only a defined and allowed set of parameters, functions and data can be exchanged via this interface, and
- if both parts cannot exchange information via any other link.

Software interfaces are part of the metrologically significant software. Circumventing the protective interface by the user is considered as a criminal action.

Acceptable solution:

> Definition of all functions, commands, data, etc., which are exchanged via the protective interface from the metrologically significant software to all other connected software or hardware parts. Checking whether all functions, commands and data are allowed.

c. Metrologically significant software shall be identified as such and shall be secured. Its identification shall be easily provided by the device for metrological controls or inspections.

This requirement implies that:

The operating system or similar auxiliary standard software, such as video drivers, printer drivers or hard disk drivers, need not be included in the software identification.

Acceptable solution:

Calculation of a checksum over the machine code of the metrologically significant software at runtime and indication on manual command. This checksum represents the metrologically significant software and can be compared to the checksum defined at type approval.

d. In addition to all other required documentation, the special software documentation shall include:

- A description of the system hardware, e.g., block diagram, type of computer(s), type of network, if not described in the operating manual (see also Table ZZ)
- A description of the software environment for the metrologically significant software, e.g., the operating system, required drivers, etc.
- A description of all metrologically significant software functions, metrologically significant parameters, switches and keys that determine the functionality of the instrument, including a declaration of the completeness of this description
- A description of the relevant measuring algorithms (e.g., stable equilibrium, price calculation, rounding algorithms)
- A description of the relevant menus and dialogues
- The securing measures foreseen (e.g., checksum, signature, audit trail)
- The complete set of commands and parameters – including a short description of each command and parameter – that can be exchanged between the metrologically significant software and the associated software via the protective software interface, including a declaration of the completeness of the list
- The software identification foreseen for the metrologically significant software
- If downloading of software via modem or internet is foreseen: a detailed description of the loading procedure and the securing measures against accidental or intentional changes
- If downloading of software via modem or internet is not foreseen: a description of the measures taken to prevent inadmissible uploading of metrologically significant software
- In case of long-term storage or transmission of data via networks: a description of the data sets and protection measures (see 5.5.3)

YY.3. Data storage devices (DSD).
If there is a device, whether incorporated in the instrument or being part of the instrument as software solution or connected to it externally, that is intended to be used for long-term storage of weighing data (in the sense of T.2.8.5), the following additional requirements apply.

YY.3.1. The DSD must have a storage capacity which is sufficient for the intended purpose.
Note: The regulation concerning the minimum duration for keeping information is outside the requirements concerning instruments and probably left to national rules concerning trade. It is the responsibility of the owner of the instrument to have an instrument that has sufficient capacity of storage to fulfill the requirements applicable to his activity. At type examination it will only be checked that the data are stored and given back correctly, and that there are adequate means foreseen to prevent the loss of data if the storage capacity is exhausted before the duration foreseen.

YY.3.2. The metrologically significant data stored must include all relevant information necessary to reconstruct an earlier weighing.
Note: Metrologically significant data are (see also T.2.8.1): gross or net values and tare values (if applicable, together with a distinction of tare and preset tare), the decimal sign(s), the unit(s) (may be encoded), the

identification of the data stored, the identification number of the instrument or load receptor if several instruments or load receptors are connected to the data storage device, and a checksum or other signature of the data stored.

YY.3.3. The metrologically significant data stored shall be adequately protected against accidental or intentional changes.
Examples of acceptable solutions:

a. A simple parity check is considered sufficient in order to protect the data against accidental changes during transmission.

b. The data storage device may be realized as an external software-controlled device using, for instance, the hard disk of a PC as the storage medium. In this case the respective software shall meet the software requirements in 5.5.2.2. If the stored data are either encrypted or secured by a signature (at least 2 bytes, e.g., a CRC-16 checksum with hidden polynomial) this will be considered sufficient in order to protect the data against intentional changes.

YY.3.4. The metrologically significant data stored shall be capable of being identified and displayed, where the identification number(s) shall be stored for later use and recorded on the official transaction medium. In case of a printout the identification number(s) shall be printed.
Example of an acceptable solution:

The identification may be realized as consecutive numbers or as the respective date and time (mm:dd:hh:mm:ss) of the transaction.

YY.3.5. The metrologically significant data shall be stored automatically.
Note: This requirement means that the storing function must not depend on the decision of the operating person. It is accepted, however, if intermediate weighings that are not used for the transaction are not stored.

YY.3.6. Stored metrologically significant data sets which are to be verified by means of the identification must be displayed or printed on a device subject to legal control.

YY.3.7. Data Storage Devices are identified as a feature, option, or parameter on an NTEP CC if they are incorporated in the instrument or form part of the instrument as software solution.

October Meeting Discussion:
The Sector agreed that Handbook 44 already has audit trail and physical seal, but these may need to be enhanced.

From WELMEC document:

Protection against accidental or unintentional changes.
Metrologically significant software and measurement data shall be protected against accidental or unintentional changes.

Specifying Notes:
Possible reasons for accidental changes and faults are: unpredictable physical influences, effects caused by user functions and residual defects of the software even though state-of-the-art development techniques have been applied.

This requirement includes:

a. Physical influences: Stored measurement data shall be protected against corruption or deletion when a fault occurs or, alternatively, the fault shall be detectable.
b. User functions: Confirmation shall be demanded before deleting or changing data.
c. Software defects: Appropriate measures shall be taken to protect data from unintentional changes that could occur through incorrect program design or programming errors, e.g., plausibility checks.

Required Documentation:
The documentation should show the measures that have been taken to protect the software and data against unintentional changes.

Example of an Acceptable Solution:

- The accidental modification of software and measurement data may be checked by calculating a checksum over the relevant parts, comparing it with the nominal value and stopping if anything has been modified.
- Measurement data are not deleted without prior authorization, e.g., a dialogue statement or window asking for confirmation of deletion.
- For fault detection see also Extension I.

Proposed checklist for Publication 14 numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussion at October Sector Meeting.

Devices with Embedded Software TYPE P (built-for-purpose)			
	Declaration of the manufacturer that the software- is used in a fixed hardware and software environment, and		**Yes** □ **No** □ **N/A** □
	cannot be modified or uploaded by any means after securing/verification		**Yes** □ **No** □ **N/A** □
	Note: *It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.*		
	The software documentation contains:		
		description of the metrologically significant functions	**Yes** □ **No** □ **N/A** □
		description of the securing means (evidence of an intervention)	**Yes** □ **No** □ **N/A** □
		software identification	**Yes** □ **No** □ **N/A** □
		description of how to check the actual software identification	**Yes** □ **No** □ **N/A** □
	The software identification is:		
		clearly assigned to the metrologically significant software and functions	**Yes** □ **No** □ **N/A** □
		provided by the device as documented	**Yes** □ **No** □ **N/A** □
Personal Computers, Instruments with PC Components, and Other Instruments, Devices, Modules, and Elements with Programmable or Loadable Metrologically Significant Software TYPE U (not built-for-purpose)			
	The *metrologically significant* software is:		
		documented with all relevant information	**Yes** □ **No** □ **N/A** □
		protected against accidental or intentional changes	**Yes** □ **No** □ **N/A** □
		Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (means of security)	**Yes** □ **No** □ **N/A** □
Software with Closed Shell (no access to the operating system and/or programs possible for the user)			
	Check whether there is a complete set of commands (e.g., function keys or commands via external interfaces) supplied and accompanied by short descriptions		**Yes** □ **No** □ **N/A** □
	Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands		**Yes** □ **No** □ **N/A** □
Operating System and/or Program(s) Accessible for the User:			
	Check whether a checksum or equivalent signature is generated over the machine code of the metrologically significant software (program module(s) subject to legal control and type-specific parameters)		**Yes** □ **No** □ **N/A** □

	Check whether the metrologically significant software will detect and act upon any unauthorized alteration of the metrologically significant software using simple software tools, e.g., text editor.		**Yes** □ **No** □ **N/A** □
Software Interface(s)			
	Verify the manufacturer has documented:		
		the program modules of the metrologically significant software are defined and separated	**Yes** □ **No** □ **N/A** □
		the protective software interface itself is part of the metrologically significant software	**Yes** □ **No** □ **N/A** □
		the *functions* of the metrologically significant software that can be accessed via the protective software interface	**Yes** □ **No** □ **N/A** □
		the *parameters* that may be exchanged via the protective software interface are defined	**Yes** □ **No** □ **N/A** □
		the description of the functions and parameters are conclusive and complete	**Yes** □ **No** □ **N/A** □
		there are software interface instructions for the third party (external) application programmer.	**Yes** □ **No** □ **N/A** □

From previous notes this may be part of another section in the publication.

Software Identification			
	The metrologically significant software is identified by a software identification		**Yes** □ **No** □ **N/A** □
	The software identification:		
		covers all program modules of the metrologically significant software and the type-specific parameters at runtime of the instrument	**Yes** □ **No** □ **N/A** □
		is easily provided by the instrument	**Yes** □ **No** □ **N/A** □
		can be compared with the reference identification fixed at type approval	**Yes** □ **No** □ **N/A** □
		spot checks whether the checksums (signatures) are generated and work as documented	**Yes** □ **No** □ **N/A** □
		there exists an effective audit trail	**Yes** □ **No** □ **N/A** □

Data Storage Devices (DSD)
From the previous meeting, this was tabled (This checklist was not reworked at this time)

5.5.3	G.3.1	DSD realised with embedded software (examine software acc. to G.1)			Yes □ No □
		DSD realised with programmable/loadable software (examine software acc. to G.1)			Yes □ No □
		documentation with all relevant information			Yes □ No □
5.5.3.1	**G.3.2**	sufficient storage capacity for the intended purpose			
		data are stored and given back correctly			
		sufficient description of measures to prevent data loss			
5.5.3.2	**G.3.3**	storage of all relevant information necessary to reconstruct an earlier weighing, i.e., gross, net, tare values, decimal signs, units, identifications of the data set, instrument number, load receptor, (if applicable), checksum/signature of the data set stored.			
5.5.3.3	**G.3.4**	protection of the stored metrologically significant data against accidental or intentional changes			
		protection of the stored metrologically significant data at least with a parity check during transmission to the storage device			
		protection of the stored metrologically significant data at least with a parity check of a storage device with embedded software (5.5.1)			
		protection of the stored metrologically significant data by an adequate checksum or of a storage device with programmable or loadable software (5.5.2)			
5.5.3.4	**G.3.5**	identification and indication of the stored metrologically significant data with an identification number			
		record of the identification number on the official transaction medium, i.e., on the print-out			
5.5.3.5	**G.3.6**	automatic storage of the metrologically significant data			
5.5.3.6	**G.3.7**	a device subject to legal control prints or displays the stored metrologically significant data for verifying			

6. Software Maintenance and Reconfiguration

After the software is completed, what do the manufacturers use to secure their software?

Source: NTETC Software Sector

From Previous Meeting:
Traced means audit trail record – requires Category 3 audit trail.

Verified means evaluator verified – requires breaking a seal and placing back into service by registered agent or W&M official. (D-SW requires agent to be present to verify the update.) It was noted that in some jurisdiction, this role may be performed by a registered service agent.

October Meeting discussion:

(This section taken from Document OIML D-SW Working Draft 1 WD and provided as background.)

Maintenance and re-configuration
Only versions of metrologically significant software that conform with the approved type are allowed for use.

Verified update

The software to be updated can be loaded locally (e.g., directly) on the weighing or measuring device or remotely via a network. Loading and installation may be two different steps (as shown in Fig. 5.1) or combined to one, depending on the needs of the technical solution. After update of the metrologically significant software of a weighing or measuring device (exchange with another approved version or re-installation), the weighing or measuring device is not allowed to be used for legal purposes before a (subsequent) verification of the instrument has been performed and the securing means have been renewed. A person responsible for verification must be at place. (NOTE: This may need to be in the HB under user requirement.)

Traced update

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

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

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

Technical means shall be employed to guarantee the authenticity of the loaded software i.e., that it originates from the owner of the type approval certificate. This can be accomplished, e.g., by cryptographic means like signing. The signature is checked during loading. If the loaded software fails this test, the instrument shall discard it and use the previous version of the software **<u>or become inoperative</u>**.

Technical means shall be employed to guarantee the integrity of the loaded software, i.e., that it has not been inadmissibly changed before loading. This can be accomplished by adding a checksum or hash code of the loaded software and verifying it during the loading procedure. If the loaded software fails this test, the instrument shall discard it and use the previous version of the software **<u>or become inoperative.</u>**

It shall be guaranteed by technical means that software may only be updated with the explicit consent of the user or owner of the measuring instrument.

If the requirements above cannot be fulfilled, it is still possible to update the legally non-relevant software part. In this case the following requirements shall be met:

- There is a distinct separation between the metrologically significant and non-relevant software.
- The whole metrologically significant software part cannot be updated without breaking a seal.
- It is stated in the type approval certificate that updating of the legally non-relevant part is acceptable.

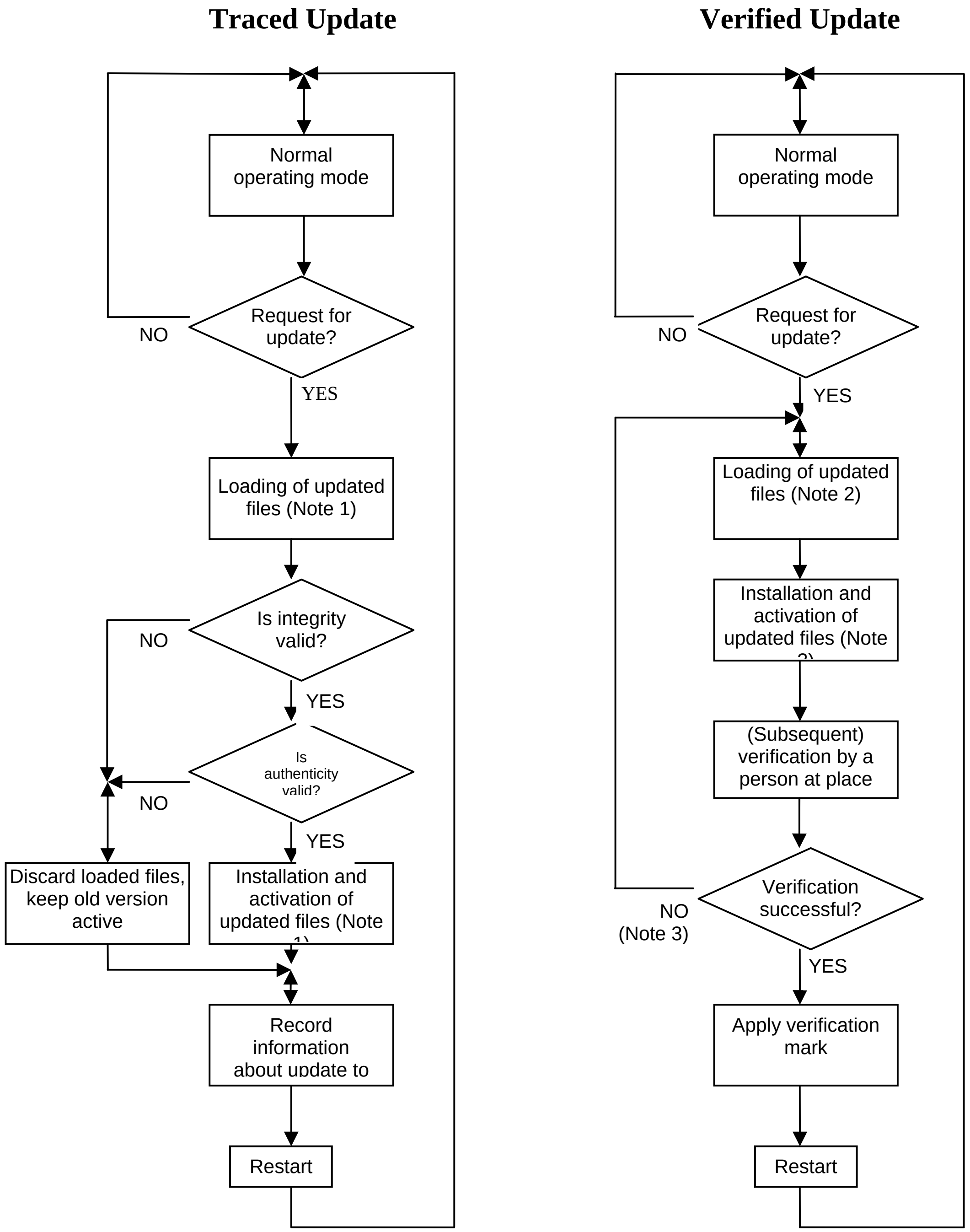
Traced Update
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 1)
Is integrity valid?
NO
YES
Is authenticity valid?
NO
YES
Discard loaded files, keep old version active
Installation and activation of updated files (Note 1)
Record information about update to
Restart
Verified Update
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 2)
Installation and activation of updated files (Note 2)
(Subsequent) verification by a person at place
Verification successful?
NO (Note 3)
YES
Apply verification mark
Restart

Figure 5-1: Software Update Procedures

Notes to

Figure 5-1:

1. In case of *Traced update*, updating is separated into the steps: "loading" and "installing/activating". This implies that the software is temporarily stored after loading without being activated because it must be possible to discard the loaded software and fall back to the old version, if the checks fail **or become inoperative.**
2. In case of *Verified update,* the software may also be loaded and temporarily stored before installation but depending on the technical solution loading and installation may also be accomplished in one step.

3. Here only failing of the verification because of the software update is considered. Failing because of other reasons doesn't require re-loading and re-installing of the software, symbolised by the NO-branch.

End of background information

Conclusions from October meeting discussion:
These four items are the accepted checklist questions:

1. Verify that the update process is documented
2. Software to be installed is authenticated and checked for integrity
3. Verify that the sealing requirements are met
4. Verify that if the upgrade process fails, the device is inoperable or the original software is restored

The manufacturer shall ensure by appropriate technical means (e.g., an audit trail) that traced updates of metrologically significant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection. This requirement enables inspection authorities, which are responsible for the metrological surveillance of legally controlled instruments, to back-trace traced updates of metrologically significant software over an adequate period of time (that depends on national legislation).

An entry is generated for each update.
The audit trail shall contain the following information:

- notification of the update procedure,
- software identification of the installed version,
- time stamp of the event,
- identification of the downloading party.

The traceability means and records are part of the metrologically significant software and should be protected as such. The software used for displaying the audit trail belongs to the fixed metrologically significant software. ***Note: This needs to be discussed further due to some manufacturers' concerns about where the software that displays the audit trail information is located and who has access if this feature is provided.***

The Sector will continue to develop this item.

7. Verification in the Field, by the W&M Inspector

Source: NTETC Software Sector

October Meeting Comments:
Question: What tools does the field inspector need?

Possible Answers:

- Have NTEP CC No. continuously displayed (needs some type of protection) during the normal weighing or measuring operation
- Clear and simple instructions on NTEP CC to get to the other Inspection Information
- The CRC, checksum, version no. etc., needs to be easily accessible from operator console.
- How to access audit trail
- System information is easily accessible (RAM, OS, etc.)
- System parameters are easily accessible (AZT, motion, time outs, etc.)

Conclusion from the October meeting:
The Sector will continue to develop this item.

8. NTEP Application

Source: NTETC Software Sector

Conclusion from the October meeting:
No direct discussion on this item took place at the October 2007 meeting.

New Items

9. Next Meeting

Conclusion from the October meeting:
The next meeting could be scheduled in conjunction with the NTEP Lab Meeting which is planned for Ottawa, Canada toward the end of April. Information regarding dates and location is now being gathered. The Sector will be notified as soon as additional information is available.