Appendix D

National Type Evaluation Technical Committee
Software Sector

May 20 - 21, 2008 – Reynoldsburg, Ohio
Meeting Summary

Agenda Items

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, Tolerances, and other Technical Requirements for Weighing and Measuring Devices*, 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 weights and measures 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.

Recommendation: There should 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 Certificates of Conformances (CC) for Software

Source: NCWM Reports

Background: Excerpts of reports from the 1995 - 1998 Executive Committees 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, Point of Sale (POS), and batch controller software) and, in some cases, had issued CCs for stand-alone software. The Board established a software work group (WG) to study the issues and make recommendations.
>
> The WG discussed many issues including: first indication of the final quantity, metrologically significant software, definitions, software marking, software checklist evaluation, a software Examination Procedure Outline (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 WG. In 1997 after the Annual Meeting, the NCWM chair appointed a new Software Work Group.
>
> **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 CC 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: The recommendation below was discussed. It was pointed out 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. The Sector 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. The Sector requests the NTEP Committee place this issue on their agenda.

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

1.c. Definitions for Software Based Devices

Source: NTETC Software Sector

Background: Discussed was marking and G-S.1.1. Location of Marking Information for Not-Built-for-Purpose, Software-Based Devices. 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. Handbook 44 does not have a definition for a not-built-for-purpose device. 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

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 an item and be considered for inclusion in NIST Handbook 44.

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). 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

(b) **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.

From NCWM Publication 16, 2008:

310-2 D Appendix D – Definition of Electronic Devices, Software-Based

Source: National Type Evaluation Technical Committee (NTETC) – Software Sector (This item was assigned developing status and moved to 360-2 Part 1, Item 2.)

Appendix A Part 1, Item 2 Appendix D – Definition of Electronic Devices, Software-Based
(This item first appeared on the 2008 S&T Committee Interim Agenda as Item 310-2)

Source: National Type Evaluation Technical Committee (NTETC) – Software Sector

Recommendation: Add a new definition and cross-reference term to Appendix D in HB 44 for "Electronic devices, software-based" as follows:

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), 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

(b) 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.

Software-based devices – See Electronic devices, software-based.

Background/Discussion: During the NTETC Software Sector discussion on marking requirements and G-S.1.1. Location of Identification Information, it was initially suggested that the term "not-built-for-purpose" be removed from the wording in NIST HB 44 paragraph G-S.1.1. since there is no definition for a not-built-for-purpose device in HB 44. After a lengthy discussion related to the terms "built-for-purpose" and "not-built-for-purpose," the Sector agreed these terms were not clear and should be replaced with the terminology proposed above. The proposed definitions are based on the revision of OIML R 76 Non-automatic weighing instruments Subsections 5.5.1. (Type P) and 5.5.2. (Type U).

At the 2008 Interim Meeting, the SMA supported the intent of the item but stated that it is premature to place these definitions in HB 44. The SMA recommended that the status of the item be changed to Developing on the S&T Committee agenda. The Committee agreed to move Item 310-2 of the 2008 S&T Committee Interim agenda and assign Developing status as 360-2 Part 1, Item 2.

Conclusion: The Sector discussed why this item was moved to Developing by the S&T Committee. It seems that the only issue in question was the use of the "aka." The Sector noted that it believes this item was already developed and should be placed on Informational status by the S&T so that additional discussion can be held on this item at open hearings.

The Sector again discussed "first final" and what is required. The NCWM Publication 14 states that first final is up to the first final indicated or recorded representation on which the transaction is based. NTEP only provides the guidelines for evaluation; it does not set regulations.

1.d. Software Identification/Markings

Source: NTETC Software Sector

Background/Discussion: During their October 2007 meeting, the Sector discussed the value and merits of required markings for software. This included the possible differences in some types of devices and marking requirements. After hearing several proposals, the Sector agreed to the following technical requirements applicable to the marking of software:

1. the NTEP CC Number must be continuously displayed or hard marked,
2. the version must be software-generated and shall not be hard marked,
3. the version is required for embedded (Type P) software,
4. printing the required identification information can be an option,
5. command or operator action can be considered as an option in lieu of a continuous display of the required information, and
6. devices with Type P (embedded) software must display or hard mark make, model, S.N. to comply with G-S.1. Identification.

The Sector developed marking information requirements and submitted a proposal to the S&T Committee for considered inclusion in NIST Handbook 44. Unfortunately, some changes made to the table as the item was prepared for Publication 16, did not reflect the content of the table as it was submitted by the Sector.

The table **as seen** in NCWM Publication 16 2008 Agenda Item:

Appendix A Part 1, Item 1 General Code: G-S.1. Identification – (Software)

Source: National Type Evaluation Technical Committee – Software Sector

Recommendation: Amend G-S.1. and/or G-S.1.1. to include the following:

Method	NTEP CC No.	Make/Model/Serial No.	Software Version/Revision[1]
TYPE P electronic devices shall meet at least one of the methods in each column:			
Hard-Marked	X	X	Not Acceptable
Continuously Displayed	X	X	X
By command or operator action	Not Acceptable	Not Acceptable	X[2]
TYPE U electronic devices shall meet at least one of the methods in each column:			
Hard-Marked	X[3]	X	Not Acceptable
Continuously Displayed	X	X	X
Via Menu (display) or Print Option	Not Acceptable	X[4]	X[4]

[1] If the manufacturer 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 Positive Displacement (P.D.) meter with integral correction, digital load cell (only for reference, not limiting).
[2] Information on how to obtain the Version/Revision shall be included on the NTEP CC.
[3] Only if no means of displaying this information is available.
[4] Information on how to obtain Make/Model, Version/Revision shall be included on the NTEP CC.

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.

The Sector reviewed this table and made both corrections and further clarifications. The table as **currently proposed** by the Sector to the S&T Committee is as follows:

The table is split into Type P and Type U devices for clarity. While there are similarities between the Type P and Type U devices, they are unique and must be treated separately.

Method	NTEP CC No.	Make/Model/Serial No.	Software Version/Revision[1]
TYPE P electronic devices shall meet at least one of the methods in each column:			
Hard-Marked	X	X	Not Acceptable[1]
Continuously Displayed	X	X	X
By command or operator action	Not Acceptable	Not Acceptable	X[2]

[1] If the manufacturer 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.~~ **the version/revision shall be hard marked on the device**. Example: Primary sensing element may be Positive Displacement (P.D.) meter with integral correction, digital load cell (only for reference, not limiting).

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

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.

Method	NTEP CC No.	Make/Model/~~Serial No.~~	Software Version/Revision
TYPE U electronic devices shall meet at least one of the methods in each column:			
Hard-Marked	X[3]	X	Not Acceptable
Continuously Displayed	X	X	X
Via Menu (display) or Print Option	Not Acceptable	X[4]	X[4]

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

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

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.

Conclusion: Submitted to NCWM S&T Committee.

2. Identification of Certified Software

Source: NTETC Software Sector

Discussion from Previous Meetings: The Sector agreed that the title of this item needs changed to "Identification of Certified Software."

- Currently, use 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 do tie the ID to the software; these include:

  Possible methods: (not limited to)
  CRC (cyclical redundancy check),
  Checksum,
  Inextricably Linked version no.,
  Encryption, and
  Digital Signature.

The question remains: Is there some method to give the weights and measures 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 (programmes, 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)

May 2008 Meeting Discussion: The Sector discussed this item at great length. The following discussion points are suggestions under consideration by the Sector:

CC would have list of functions.

One suggestion is to have the manufacturer have "some number" that is "inextricably linked" to the software version; one method is CRC.

There is the suggestion that information will be on the CC as to how the inspector can find the information on the "device" regarding the software version or other methods of identification.

It seems the software developers in attendance do not have a problem with putting a statement in Publication 14 that if you have a CC, you have a version number. The inspector then can have a means of tying the version number that he/she sees when they walk up to the device to the information on the CC. The method to do this will be defined by the manufacturer and will be verified by the NTEP lab during evaluation of the device. The list of CRC, digital signature, inextricably linked, checksum are some possible methods to do this.

Question: Is the checksum or CRC on the CC? There was a response that there needs to be information on the CC that would indicate the CRC or checksum, etc. One possibility is an "audit trail" of changes that is on the device.

Fees may be an issue, but that does not need to be considered at this point.

Timing and lab backlog must also be considered.

In WELMEC, every change is reported, and they decide what is significant or not.

In discussion on tare values, is there a need to ID the tares with a checksum? This seems to be too extreme, this is auditable data. This must be accessed; this is like a unit price on a gas pump. Tare data is not included in the metrologically significant software part!

A member stated perhaps there should only be one "metrologically significant software part" if we use the same terminology as the international community, hence the change in plurality here.

How does a field inspector verify the proper tare was used if someone complains about a transaction a few days afterward (or a series of transactions)? Perhaps the tare data is being stored externally (e.g., a central host), so another question is how do you enforce proper Category III logging in a distributed system like that?

Example from DSW 2 CD:

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of the checksum as determined by algorithm **XYZ** is **1A2B3C**.
>
> Possibly "parametric data" could be used.

The Sector discussed the definition of an "enclosed system." This means that the manufacturer has compiled their own software, and it is distributed to their own facilities or it runs on a server at a main location. There is "limited" access to the software from outside the "circle."

Conclusion: The item needs additional discussion and development by the Sector.

3. Software Protection/Security

Source: NTETC Software Sector

Background from Previous Meetings: The Sector agreed that Handbook 44 already has audit trail and physical seal, but these may need to be enhanced.

From the 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.

The Sector continued to develop a proposed checklist for Publication 14. The numbering will still need to be added. This is roughly based on R 76-2 checklist and discussion from the October 2007 Sector meeting.

The NTEP labs have been asked by the Sector Chair to begin to use this checklist for new devices coming into the labs. The main purpose of this trial by the NTEP labs is to begin to gather information on any possible problems with the checklist. At this point, this is a draft only and has not been submitted for review by the NTEP Committee.

The information requested by this checklist is currently voluntary; however, it is recommended that applicants comply with these requests or provide specific information as to why they may not be able to comply. Based on this information, the checklist may be amended to better fit with NTEP's need for information and the applicant's ability to comply.

The California, Maryland, and Ohio labs agreed to use this checklist on one of the next devices they have in the lab and report back to the Sector on what the problems may be.

Devices with embedded software TYPE P (aka 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 (all) metrologically significant functions (OIML states that there shall be no undocumented 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 (aka not-built-for-purpose)			
	The *metrologically significant* software is:		
		documented with all relevant (see below for list of documents) 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 (e.g., physical seal, checksum, CRC, audit trail, etc., 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~~ W&M jurisdiction 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 OIML DSW-2 CD as a reference ONLY.

x.y.z. Typical **Required** Documentation (for each measuring instrument, electronic device, or sub-assembly) basically includes:

- A description of the ~~legally relevant~~ metrologically significant software and how the requirements are met;
  - List of software modules that belong to metrologically significant part ~~(Annex B)~~ including a declaration that all metrologically significant functions are included in the description;
  - Description of the software interfaces of the metrologically significant software part and of the commands and data flows via this interface including a statement of completeness ~~(Annex B)~~;
  - Description of the generation of the software identification;
  - ~~Depending on the validation method chosen in the relevant OIML Recommendation (see 6.4) the source code shall be made available to the testing authority if high conformity or strong protection is required by the relevant OIML Recommendation;~~
  - List of parameters to be protected and description of protection means;
- A description of suitable system configuration and minimal required resources (see 5.2.4);
- A description of security means of the operating system (password, … if applicable); (who controls the system, and at what level);
- A description of the (software) sealing method(s) (what may be altered, and how to keep from being altered);
- An overview of the system hardware, e.g., topology block diagram, type of computer(s), type of network etc. Where a hardware component is deemed legally relevant metrologically significant (find and replace) or performs metrologically significant functions, this should also be identified;

- A description of the accuracy of the algorithms (like filtering of A/D conversion results, price calculation, rounding algorithms, …);
- A description of the user interface, menus and dialogues;
- The software identification and instructions for obtaining it from an instrument in use;
- List of commands of each hardware interface of the measuring instrument/electronic device/sub-assembly including a statement of completeness;
- List of durability errors that are detected by the software and if necessary for understanding, a description of the detecting algorithms (we may not understand this one);
- A description of datasets stored or transmitted;
- If fault detection is realised in software, a list of faults that are detected and a description of the detecting algorithm;
- An overview of the system hardware, e.g., topology block diagram, type of computer(s), type of network etc.;
- The operating manual.

This will go under a heading and be placed in a documentation paragraph.

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 check whether the ~~checksums (signatures) are generated and~~ means of identifying the software works as documented		**Yes** ☐ **No** ☐ **N/A** ☐

	~~The audit trail **(this needs to be changed to reflect a software update log)** shall update and display (show, indicate) when the software version has changed~~ ~~An entry is generated for each software update.~~ ~~The software log/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.~~ Updates to software shall be either manually verified (Verified Update) or automatically performed and traced (Traced Update). For a Traced Update, an event logger is required. An entry shall be generated for each software update and must include the following: • an event logger (with a minimum of 10 updates), • the parameter ID, which indicates the software update, • the date and time of the change, and • the new value of the parameter, which is the software identification of the installed version.	**Yes □ No □ N/A □**

This information may need to be included in HB 44. It may be possible to add this to the General Code section.

May need to define what a software update log is.

G-S.9. Verification of Software Update

Only versions of metrologically significant software that conform to the approved type are allowed for use.

Updates to software shall be either manually verified (Verified Update) or automatically performed and traced (Traced Update).

For a Traced Update, an event logger is required. An entry shall be generated for each software update and must include the following:

- an event logger (with a minimum of 10 updates),
- the parameter ID, which indicates the software update,
- the date and time of the change, and
- the new value of the parameter, which is the software identification of the installed version.

~~An entry is generated for each software update.~~
~~The software log/audit trail shall contain the following information:~~
- ~~parameter ID; software update, etc,~~
- ~~new value; software identification of the installed version,~~
- ~~date and time of the change,~~
- ~~identification of the downloading party. (considered this~~

~~*The device shall clearly indicate that it is in the remote configuration mode and record such message if capable of printing in this mode or shall not operate while in this mode.*~~

If the device continues to operate during a software update, then the metrological performance shall not be affected.

The Maryland lab wanted it on record that they disagree with this statement and are striking the first sentence based on discussions within the Weighing Sector and the Measuring Sector and the NTEP lab meetings on the subject of calibration and configuration while in the normal weighing measuring mode. The sentence that has been struck out was placed in the DES checklist years ago to address field concerns.

It was noted there is a statement in the WELMEC document that concurs with the statement above as stricken.

Use of a Category 3 audit trail is acceptable for the software update logger.

Definitions Recommendation:

> **Verified Update.** A verified update is the process of installing new software where the security is broken and the device must be re-verified. Checking for authenticity and integrity is the responsibility of the owner/user.
>
> **Traced Update.** A traced update is the process of installing new software where the software is automatically checked for authenticity and integrity, and the update is recorded in a software update log or audit trail.

Note: The Sector agreed that these two definitions directly above for Verified Update and Traced Update were acceptable.

Question: Do we need the definitions below any longer? Comment: There is text in these definitions that doesn't belong in the definition, but may be applicable for other purposes, primarily the bit about the software protection environment being at the same level after upgrade when doing traced update. The Sector has not addressed that yet and it is important.

Previous definitions:

> **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 combined to one, depending on the needs of the technical solution. After update of the metrologically significant software of a weighing or measuring device, exchanged 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 has been renewed. A person responsible for verification must be at place. (**NOTE:** This may need to be in the handbook under user requirement.)
>
> **Traced Update.** Traced update is the procedure of changing software in a weighing or measuring 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 a software log or audit trail.
>
> 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.

Comment: The data storage device does not appear to be appropriate for the U.S. weights and measures system.

A member provided an explanation of a Data Storage Device (DSD) explaining it is an EU requirement for "legal requirements." This is the alibi memory that is a replacement for the paper printout that is required in EU. A Watt Meter will also act as DSD and store information on electricity usage over a long period of time.

The Sector agreed to delete the DSD checklist from future discussions of this Sector.

~~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~~			
~~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~~			

Conclusion: The Sector agreed to further develop a proposal to forward to the S&T Committee, adding a Section G-S.9. and two definitions to Handbook 44. It was agreed the Item G-S.9. would be sent out for ballot to the Sector members and meeting attendees.

[**Note:** In the summer of 2008, a ballot was sent to all members of the Sector. A majority of the members returning ballots voted in favor of the proposal (7 to 2). However, there were several comments received from both yea and nay voters regarding the proposal. After review of the comments, the Sector Chair decided that, considering all the circumstances, the Sector needed more discussion on the item before it is moved forward in the process and is submitted to the S&T Committee.]

4. Software Maintenance and Reconfiguration

Source: NTETC Software Sector

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

Discussion: The following items were reviewed by the Sector. Note that agenda Item 3 also contains information on Verified and Traced updates and Software Log.

a. Verify that the update process is documented (OK)

b. For traced updates, installed software is authenticated and checked for integrity

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 either use the previous version of the software or become inoperative.

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 e.g., 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 either use the previous version of the software or become inoperative.

Examples are not limiting or exclusive.

c. Verify that the sealing requirements are met

The Sector asked, "What sealing requirements are we talking about?"

This item is only addressing the software update; it can be either verified or traced. It is possible that there are two different security means, one for protecting software updates (software log) and one for protecting the other metrological parameters (Category I, II, or III method of sealing).

Some examples provided by the Sector members include but are not limited to physical seal, software log, Category III method of sealing and can contain both means of security.

d. Verify that if the upgrade process fails, the device is inoperable or the original software is restored

The question before the group is, "Can this be made mandatory?"

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).* The statement in italics will need to be reworded to comply with U.S. weights and measures requirements.

See agenda Item 3, G-S.9.

Only versions of metrologically significant software that conform to the approved type are allowed for use.

Updates to software shall be either manually verified (Verified Update) or automatically performed and traced (Traced Update).

For a Traced Update, an event logger is required. The logger shall be capable of storing a minimum of the 10 most recent updates. An entry shall be generated for each software update and must include the following:

- the event type/parameter ID, which indicates a software update event (if not using a dedicated update log),
- the date and time of the change, and
- the new value of the parameter, which is the software identification of the newly installed version.

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 requires further discussion 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. Manufacturers did indicate that there are methods available to encrypt the audit trail information; however, it cannot be protected from being deleted.

The following flowchart is sourced from OIML TC 5/SC 2, D-SW and is currently under revision.

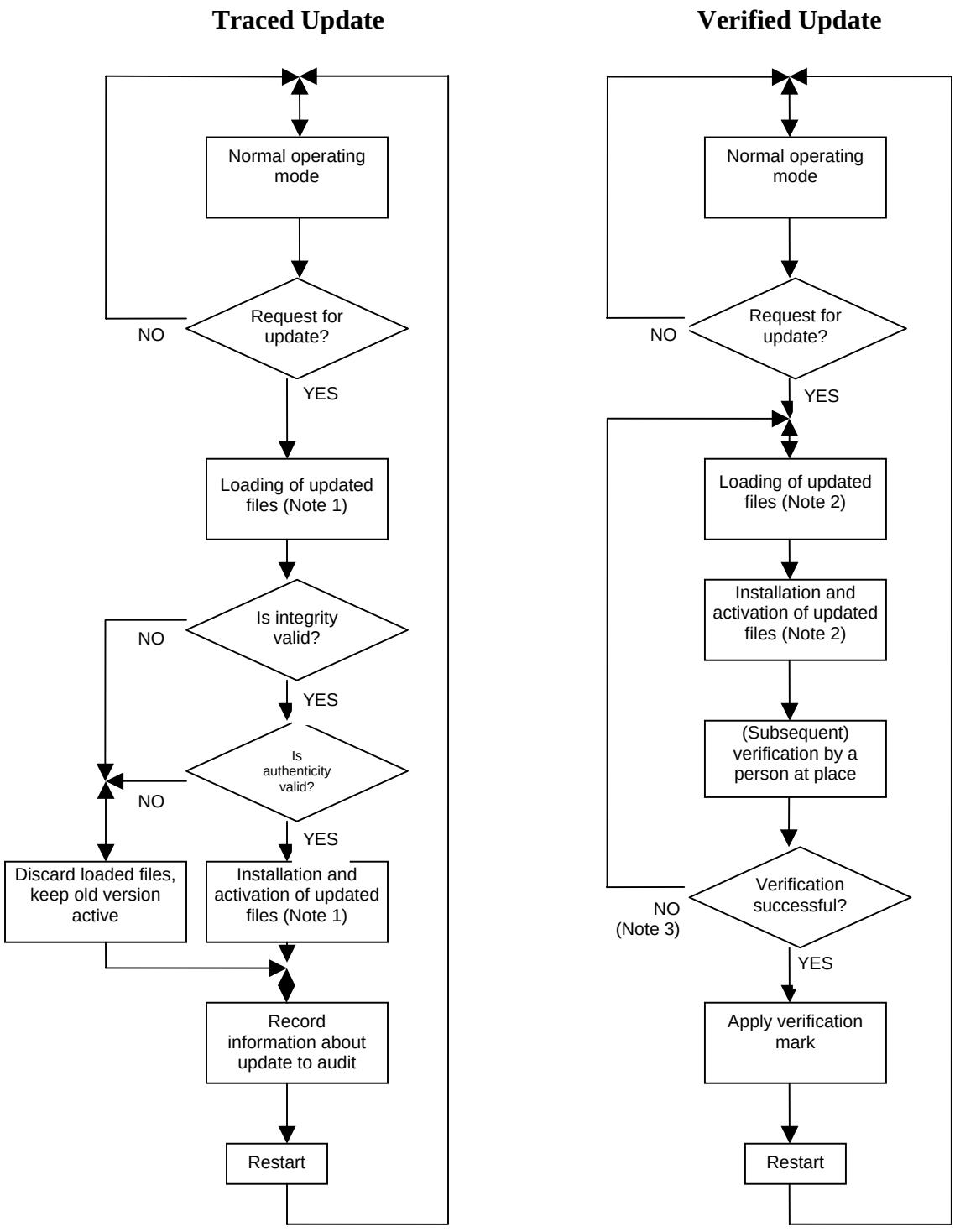

Figure 5-1: Software update procedures

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 if the checks fail, and either fall back to the old version, **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.

Conclusion: This agenda item is closely tied to agenda Item 3, Software Protection/Security; in fact much of the content from previous Sector reports has been moved to Item 3. This item 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.

5. Verification in the Field, by the Weights and Measures Inspector

Source: NTETC Software Sector

Background Question: What tools does the field inspector need?

Possible Answers:

- NTEP CC number are 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.
- CRC, checksum, version number etc., needs to be easily accessible from operator console.
- Inspector needs to know how to access audit trail.
- System information is easily accessible (ram, OS, etc).
- System parameters are easily accessible (AZT, motion, time-outs, etc.).

May 2008 Meeting: There was no additional discussion on this item. The Sector will continue to develop this item.

6. NTEP Application

Source: NTETC Software Sector

May 2008 Meeting: There was no additional discussion on this item by the Sector at this time.

New Items

7. Recommendation on Sector Chair and Technical Advisor

Source: NTEP Director

Background: With the changes to the management structure of NCWM, the Sector will need to discuss and make recommendations regarding persons to fill the roles of (NTETC) Sector Chair, and Technical Advisor to the Sector. Refer to NCWM Publication 14 Administrative Policy Section B. Administration, Subsection B.3. Paragraph 2, page AP-4.

Recommendation to NTEP Committee: The Sector discussed various options and candidates and now recommends the following Sector members for the described roles.

Documentation (scribe):	Teri Gulke, Liquid Controls
Technical Advisor:	Doug Bliss, Mettler-Toledo
Co-Sector Chairs:	Norm Ingram, California Division of Measurement Standards Jim Pettinato, FMC Technologies

8. Next Meeting

The Sector members were informed they are now on a yearly schedule for Sector meetings.

The Sector discussed the pros and cons of various meeting times and coordination with other NTEP or NCWM meetings. The NTEP Administrator will determine when the next meeting is possible.