Summary of Software Sector Meeting
Reynoldsburg, OH
May 20, 21, 2008

1a. NTETC Software Sector Mission (No additional discussion required)

no changes

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

no comments

1c. Definitions for Software Based Devices

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 that 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.

1d. Software Identification / Markings

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.

Final Summary of Software Sector Meeting May 2008

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 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.

Changes are noted in Yellow Highlights

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.

2. Identification of Certified Software

The Sector discussed this item at great length. The following items are suggestions of the Sector.

CC would have list of functions
One suggestion is to have Mfg have "some number" that is "inextricably linked" to the software version; one method is CRC

There is the suggestion that info 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.

SUGGESTION From The Sector: The developers do not have a problem with putting a statement in Pub 14 that you have a CC, you have a version no. the inspector then can have a means of tying the version no. that he/she sees when they walk up to the device and 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 info 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.

Discussion on Tare values, and the 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 unit price on a gas pump.

!!!! Tare data is not included in the metrologically significant software part !!!!

Comment: JMP: 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….

Comment: JMP: So 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?)?? If I recall the discussion, there were some possibilities like the tare data being stored externally (e.g. a central host) – so another question is how do you enforce proper Cat III logging in a distributed system like that?

Example from DSW 2CD:

The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of 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 mfg. 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".

3. Software Protection / Security

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

Question: Can labs use this check list on one of the next devices they have in the lab and report back to the group on what the problems may be? The labs agreed.

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 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-2CD 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 data sets stored or transmitted;~~
- If fault detection is realised in software, a list of faults that are detected and a description of the detecting algorithm;
- ~~An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network etc;~~
- The operating manual.

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

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

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 with 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.

(MD disagrees with this statement and striking the first sentence)
Comment: AB: based on discussions within the weighing sectors 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.

Comment: There is a statement in the WELMEC document that concurs with statement above as stricken.

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

From JIM P: definitions

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.

Comment: The **sector agreed** that the two definitions directly above for Verified update and Traced update were acceptable.

SAP Question, do we need the definitions below any longer?

Comment: JMP: There is text in these definitions that in my opinion don'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… I don't think we've addressed that yet and it is important.

Previous definitions from (_____????)

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. above) 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

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.

!!! The DSD does not appear to be appropriate for the US W&M !!!

Doug Bliss, provided an explanation of Data Storage Device, This is a EU requirement for "legal requirements" this is the alibi memory that is a replacement for the paper print out that is required in EU. A Watt Meter will also act as DSD, and store info on electricity usage over a long period of time.

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~~			

Comment on this item: AS A GROUP? Do we agree? **Yes**.

The item G-S.9. will be sent out for ballot to the sector members and meeting attendees.

4. Software Maintenance and Reconfiguration

The Following Items were reviewed by the Sector. Note that item 3 above also contains information on Verified and Traced updates and Software Log.

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

2. 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.

3. Verify that the sealing requirements are met

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

This item is <u>only</u> addressing the <u>software update</u>, 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 can contain both means of security

4. 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 US W&M requirements.

See item 3 above, G-S.9.

Only versions of metrologically significant software that conform with 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 long),
- 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 needs to be discussed further due to some manufacturer's concerns about where the software that displays the audit trail information is located and who has access if this feature is provided.***

MFG did indicate that there are methods available to encrypt the audit trail information; however, itt cannot be protected from being deleted.

The following Flow Chart is sourced from OIML TC5/SC2, D-SW and is currently under revision.

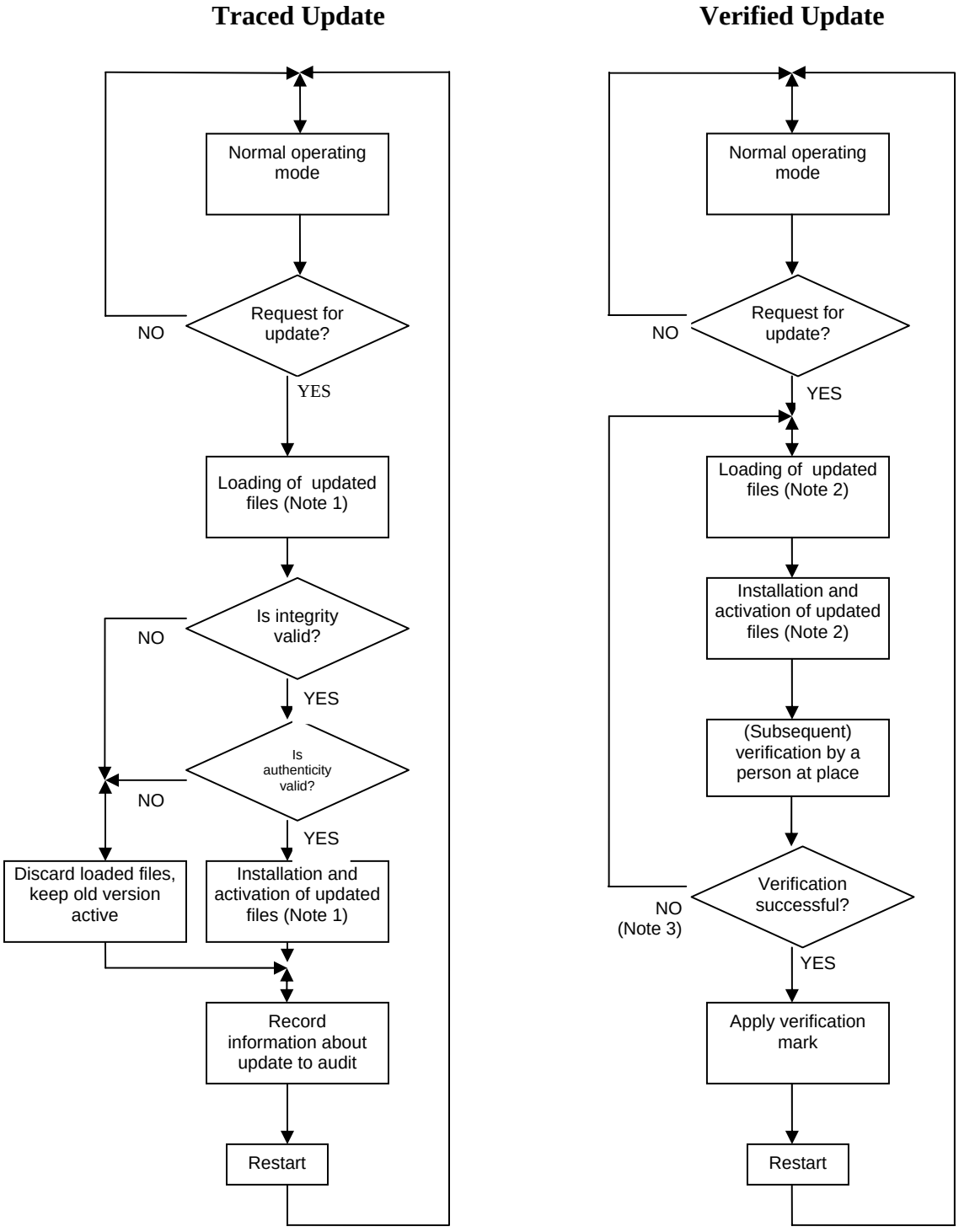

Figure 5-1: Software update procedures

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.

5. Verification in the Field, By the W&M Inspector

The Sector briefly discussed this item.

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.
- 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)

The sector will continue to develop this item.

6. NTEP Application

There was no additional discussion on this item by the Sector at this time.

7. Recommendations by the Sector on Sector Chair and Technical Advisor

The Sector discussed various options and candidates and now recommends the following Sector members for the described roles.

Documentation
Teri Gulke,

Tech Advisor
Doug Bliss,

Co-Sector Chairs

Norm Ingram
Jim Pettinato

8. Next meeting

TBD. 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.