Meeting Summary for the Second Meeting of the Software Sector
October 18-19, 2006
Annapolis, MD

Repeat of the 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.

<u>There were several general items presented by the Tech Advisor to begin the dialogue.</u>

A) During a time after the previous meeting, there was significant activity on the Software Sector listserv. However, more recently this activity has been minimal. This is a new Sector with many new members and a very difficult task. With all that we have going on it is possible that focus on the work of Software Sector was lost to our regular duties of our "real" jobs. Also, in review of the information that was sent out, it may not have been stated clearly enough what the expectations were for the working groups. To date, I have not received any reports back from the working groups. This is in no way criticism of the groups

SUMMARY: *This led to a discussion regarding previous emails. It was clear that everyone may not have received all of the emails via the listserv. It was also apparent that not all of the current interested parties were on the listserv. The NTEP staff will verify and double check that all interested parties will be on the listserv. Previous emails will be sent to the current members of the listserv.*

B) There has been a suggestion that at the start of the upcoming meeting, the groups be given a short time (30 minutes) to review what they have completed. However, there was concern that many of the people have volunteered for more than one working group. It would be difficult to be in more than one discussion at a time. Also, it would be difficult to have so many conversations going on in one room..

SUMMARY: *There was a brief discussion of each of the action items, but it became obvious that there would need to be more detailed discussion on each of the items.*

C) Another suggestion was for the group to begin the discussion by attempting to answer the question: *Should we make an effort in all that we do to harmonize with OIML, or some other current standard?* It was suggested that with such a decision, it may be more clear which direction each of the Action Items should go. It may also limit the amount of work needed to develop each of the Action Items. On the other hand, it may be the will of the group to not go in that direction due to the content or philosophy of the current OIML (or other) standard and this is not the direction that this group believes is best for evaluation, certification and verification of these types of devices under the US system of W&M.

SUMMARY: *It was clear from the discussion that not all in the room agreed with all parts of the WELMEC, FDA or OIML documents. However, it was clear that it would be useful to use as much from these documents as possible.*

Jim Truex repeated that OIML standards are international standards and we are obligated to look at these various documents and attempt to align wherever possible. He also indicated that all the documents are able to be modified when there is technical justification for such changes.

D) The previous discussion led to a discussion on "first final". Dave Vande Berg wanted to propose that this term be more clearly defined, or the definition be changed to "initial gross quantities".

Jim Truex paraphrased the current definition from NCWM Publication 14 Administrative Policy. Section C. DEVICES TO BE SUBMITTED FOR TYPE EVALUATION TYPE EVALUATION states:

> "In general, type evaluations will be conducted on all equipment that affect the measurement process or the validity or 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."

After a brief discussion, Jim Truex called for the question: Should there be a change to the definition of "First Final"

CONCLUSION:
There was an actual vote taken, with:

1 in favor of changing the definition
27 were not in favor of changing the definition, (leave it as it currently is)

Item 1. Software Identification [model/version, use of help screen, etc.?]

a. Built for Purpose
b. Not Built for Purpose
c. version number or greater

John Roach (CA), Todd Lucas (OH), Paul Lewis (Rice Lake), Bob Hoblit (IBM), Mike McGhee (Actaris), Dave Vande Berg (Vande Berg Scales), Chris Scott (Gilbarco)

SUMMARY: *There was a lengthy discussion on this item/topic. This also carries into several other agenda items. 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 was also the suggestion to use the definitions from the WELMEC document for Type P and Type U instruments. This was modified per suggestions to include the words highlighted in yellow. It was also suggested that a list of examples be provided.

DEFINITIONS:

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 is likely to contain many of the 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 legally relevant functions.

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

Then there was a discussion on specific sections of the WELMEC document that deal with TYPE P and TYPE U requirements. Comments and Question are highlighted in yellow.

P1: Documentation

In addition to the specific documentation required in each of the following requirements, the documentation shall basically include:

a. A description of the legally relevant software.

b. A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).

c. A description of the user interface, menus and dialogues.

d. The unambiguous software identification.

e. An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.

f. The operating manual.

This should not be an issue,

P2: Software identification

*The legally relevant software shall be clearly identified. An identification of the software shall be inextricably linked to the software itself. It shall be presented on command or during operation**.***

We should have addressed this in previous notes. Action item 1

P3: Influence via user interface

Commands entered via the user interface shall not inadmissibly influence the legally relevant software and measurement data.

There shall be a means to prevent changes from the user interface to any metrologically significant portions of the software and measurement data without authorization.(?) May need to define authorization per HB 44.

P4: Influence via communication interface

Commands inputted via communication interfaces of the instrument shall not inadmissibly influence the legally relevant software and measurement data.

There shall be a means to prevent changes from the communication interface to any metrologically significant portions of the software and measurement data without authorization. (?) May need to define authorization per HB 44.

P5: Protection against accidental or unintentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software and measurement data shall be protected against accidental or unintentional changes.

P6: Protection against intentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software shall be secured against the ~~inadmissible~~ unauthorized modification, loading or swapping of hardware memory.

P7: Parameter protection

~~Parameters that fix legally relevant characteristics~~ Metrologically Significant Parameters of the measuring instrument shall be secured against unauthorized modification.

U1: Documentation

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

a. *A description of the legally relevant software functions, meaning of the data, etc.*

b. *A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).*

c. *A description of the user interface, menus and dialogues.*

d. *A legal software identification.*

e. *An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.*

f. *An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.*

g. *The operating manual.*

This one should be acceptable.

U2: Software identification

The ~~legally relevant~~ (metrologically significant, [find and replace]) software shall be clearly identified. An identification of the software shall be inextricably linked to the software itself. It shall be determined and presented on command or during operation.

Inextricably (cannot be separated)
AB: put a note in the checklist for the lab that they cannot "change" the ID?
This should be covered in permanence of marking

U3: Influence via user interfaces

Commands entered via the user interface shall not inadmissibly influence legally relevant software and measurement data.

Use words from P3

U4: Influence via communication interface

Commands or other inputs via ~~non-sealed~~ communication interfaces of the device shall not inadmissibly influence the legally relevant software and measurement data.

There are question on "sealed" this may not be a physical seal.
Being a U, the person selling, may not know about all of the interfaces

There shall be a means to prevent changes from any communication interface to ~~any~~ metrologically significant portions of the software and measurement data without authorization. (?) May need to define authorization per HB 44.

Means to prevent??
The word Commands, may limit what will need to be evaluated. That may be the intent

Done for now.

U5: Protection against accidental or unintentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software and measurement data shall be protected against accidental or unintentional changes.

OK

U6: Protection against intentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software and measurement data shall be secured against ~~inadmissible~~ unauthorized modification.

OK

U7: Parameter protection
~~Legally relevant~~ (metrologically significant, [find and replace]) parameters shall be secured against unauthorized modification.

Specifying Notes:

1. Type specific parameters are identical for each specimen of the type and are in general part of the program code i.e. part of the legally relevant software. Therefore requirement U6 applies to them.

2. Device specific parameters:

"Secured" parameters may be changed using an on-board keypad or switches or via interfaces but only *before* the action of securing. Because device specific parameters could be manipulated using simple tools *on universal computers they shall not be stored in standard storages of a universal computer*. Storing of these parameters is acceptable only in additional hardware.

Settable device specific parameters may be changed after securing.

OK

U8: Software authenticity and presentation of results
Means shall be employed to ensure the authenticity of the ~~legally relevant~~ (metrologically significant, [find and replace]) software. The authenticity of the results that are presented shall be guaranteed.

Had discussion on this on Weds,
RM, there is a method to ID that this is the actual software, trace update,

It shall not be possible to fraudulently simulate approved ~~legally (MS) relevant~~ *(metrologically significant, [find and replace])* software using simple software tools.

Definition for simple software tools, e.g. text editor, notepad, office tools, and other commonly available software tools.

U9: Influence of other software
The ~~legally relevant~~ (metrologically significant, [find and replace]) software shall be designed in such a way that other software does not inadmissibly (??) influence it.

This is DOOM!

A suggestion to consider a metrological device table was presented to the group. After modifications were made, the following table was discussed.

General Marking of Metrological Devices

	Software Only (this is U)	**Software + Hardware (this is P)**	**Hardware Only (this is neither P nor U, mechanical)**
Make	X	X	X

Model	X	X	X
Revision/Version	X	X	
COC	X	X	X
Serial Number		X	X

CONCLUSION: *It is apparent a lot more study and understanding of these complex issues are necessary. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

Item 2. Software Protection/Security

2a. Identification of unapproved/unauthorized software

Mike Roach (Verifone), Norm Ingram (CA), Bob Hoblit (IBM), Andre Elle (Endress& Hauser), Rich Miller (FMC), Dave Vande Berg (Vande Berg Scale), Chris Scott (Gilbarco), Doug Bliss (Mettler-Toledo)

There is no way to tell someone how to do sealing, you can say what needs to be accomplished.
JT, segregation of parameters is currently allowed. (table of sealable parameters)

JT right now there are two methods, physical seal, audit trail, does the group believe that there needs to be some other category.

NOTES:
Norm was attempting to get input on what current protection is in place, used by manufacturer

Email chain on this item.

There is a discussion on this item.
JT asking the question: will audit trail work for "sealing" software?

Currently, industry does protect, but it is not audit trail.
There is an issue of audit trail, if the software is not running, or have a software service, the changes could be made and not tracked by audit trail.

Checksum:

DB the only way to know is by a bit by bit comparison, which is not practical.
BF uses the checksum in Taxi meters.

?? There is no way to tell someone how to do sealing, you can say what needs to be accomplished.

SW has several examples of methods of sealing.
authentication

Access control
There is also a spec for certification

X509 Certificates,
PCATS certifies vendors
Version Number, application (checksum) There is a challenge response with different certifications. They validate who they are, there may also be limits set. Receive data verification.

JT, segregation of parameters is currently allowed. (table of sealable parameters)

JT right now there are two methods, physical seal, audit trail, does the group believe that there needs to be some other category.

DB: does not believe that HB 44 does not need to be changed.

DO: Needs to know that software is not being manipulated,
SW: What are we attempting to solve?
DO: POS system not certified, made change, got CC, then, there was ability to change version on the screen, WI-WO, Manual Weight entry as a sealable parameter.
SW: X509 Certification, it is something like verisign, electronic signature and verification.
JT could someone put something together?
WS: Canada has a section

DB: Scale System Controller
The scale system controller has approval certifications for USA and the European Union. In this case, a Commercial Off The Shelf (COTS) PC is used in conjunction with a scale system (terminal and weigh platform). The scale system provides the PC with approved gross weight and accepts commands to zero the weight indication. The PC application program

- stores and recalls weights
- computes net weight using a stored weight or manually entered weight
- provides the user display of net weight
- may compute price based on the net weight and a selected commodity code
- may print a weigh ticket

Protection of configuration and price parameters
Metrologically significant parameters are maintained within the scale terminal and are controlled there. Other parameters are stored in a password protected database. The user controls password protection access and distribution.

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

Protection of software
Metrologically significant software is supplied only as binary code. Each such module is protected by a CRC32 checksum. The expected checksums, revision levels, and dates are kept in an encrypted configuration file. If run-time values differ from expected values the system will not operate. The configuration information can be recalled by an inspector using the Help/About menu in the application program.

Protection of active data
Data from the scale terminal is wholly owned by the scale server metrological interface. No other agent can acquire that data when the scale server is running, and the application program will not accept data except from the scale server.
Transactional information is stored in an encrypted Alibi Memory log. No access is permitted to this data except via the supplied application program. Data can be exported via the application program for external use, but no user modifications are permitted to the original transaction data.

Protection of operating system user interface
There are no special restrictions to the operating system. The application program runs as any other on the PC and can be started, stopped, or minimized.

JP: could you replace the word "data" to software or program.
DVB: if the code is complied, there is little chance of changing the code
DO: Facilitation of Fraud has changed since it was first put in the HB. Industry needs to define what is reasonable.
AB: can we take some of the info from WELMEC and use it?

JT: have some agreements but still not in agreement, leave

DB: In Europe, there are things like, safety, highest level security etc. First modification there would be a limit to the risk classes.

P5: Protection against accidental or unintentional changes
Legally relevant 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 of 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.

Validation Guidance: Typical Examples
Checks based on documentation:
□ Check that a checksum of the program code and the relevant parameters is generated and verified automatically.
□ Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.
□ Check that a warning is issued to the user if he is about to delete measurement data files.
Functional checks:
□ Check by practical spot checks that before deleting measurement data a warning is given, if deleting is possible at all.

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 authorisation, e.g. a dialogue statement or window asking for confirmation of deletion.
□ For fault detection see also Extension I.

CONCLUSION: *There is a need for evaluation and work on this area. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

Item 3. Storage of Data [subsections, automatic storing and transmission]

Rich Miller (FMC), Keith Harper (Gencor), Dave Hoffman (Toptech),
Mike Roach (Verifone)

After much discussion the question was raised whether NTEP should dictate how data is stored or require security and allow the manufacturer to design the means, as is done with other devices.

CONCLUSION: *Doug Bliss, motion to place item 3 on low priority and discuss at a later date. Table this indefinitely, This is not outside the realm, but choose not to look at it at this time.*

In Favor: most of the room
Oppose 2

The item will be removed from the agenda at this time but the group will have the opportunity to bring this back up at ay future meeting.

Item 4. Software Maintenance and Reconfiguration

Wayne Stiefel (NIST), Tony Herrin (Cardinal), Gary Lameris (Hobart), Rich Miller (FMC), Bob Hoblit (IBM), George Brazis (Avery), Travis Gibson (Rice Lake), Keith Harper (Gencor)

NOTES:

DB User must agree with the upgrade.
JT: Question, Storage of Data, current NTEP CC, requirements. MSC's
NI: don't we already do this (section 4)
JT does this group need to look at this?
AB: question about if an update, does it need to reconfigure the system?
WS: this should be part of the process since that you are doing the other areas
JT: How many believe that we do not have to address this

Does the sector need to address this issue?

There was a split vote,

CN: need to include the other side of the flow chart to include physical seals.

OIML D-SW 5.2.6.
Only versions of legally relevant software that conform with the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also on the kind of instrument under consideration. The following options 5.2.6.1 and 5.2.6.2 are equivalent alternatives. This issue concerns verification in the field. Refer to chapter 7 for additional constraints.

This follows the traced update, the verified update is still an option

This appears to be covered by Cat 3 and enforcement.
This may appear to be covered by other sections or security
This section should not include eproms
WS: is there a security key?

AB: does it download correctly?
AB: if they are downloading software, then it may need to be Cat 3
WS: OIML says that the audit trail needs to be updated
AB: this may be only guidance in Pub 14 and not a change to HB 44.

The following flow chart, developed to assist the manufacturer/designer was discussed in depth.

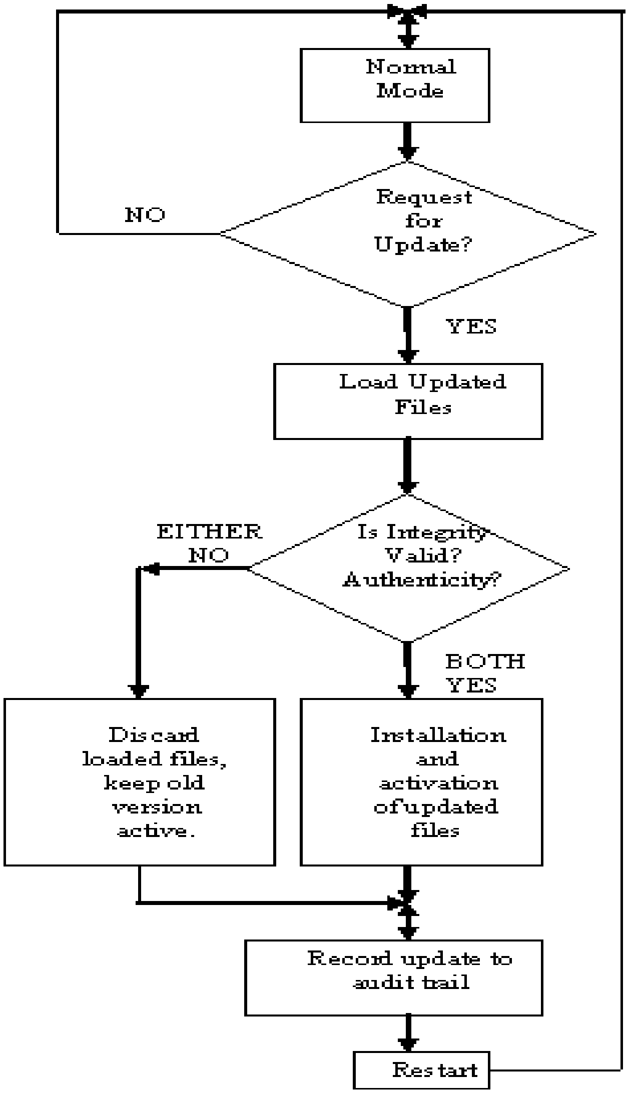

Normal Mode
Request for Update?
NO
YES
Load Updated Files
Is Integrity Valid? Authenticity?
EITHER NO
BOTH YES
Discard loaded files, keep old version active.
Installation and activation of updated files
Record update to audit trail
Restart

Figure 1.0 Traced Update Requirements

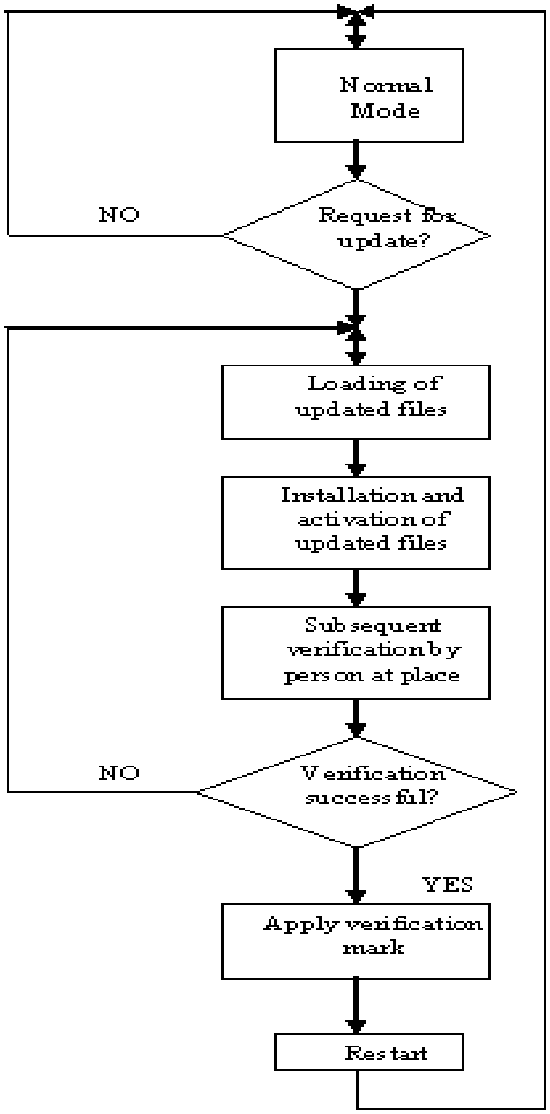


Figure 2.0 Verified Update Model

CONCLUSION: *It is apparent a lot more study and understanding of these complex issues are necessary. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

5. *V*erification in the field, needs work

Jim Truex (OH), Jim Sexton (Rice Lake), Keith Harper (Gencor)

Jim T. Field inspector will need help in the field with verification. Ohio

CONCLUSION: *Cover this at another time. Ohio has developed a field checklist that may be used as a starting point.*

6. NTEP Application – [mfg documentation to be submitted]

Steve Patoray (NTEP), Paul Lewis (Rice Lake), Keith Harper (Gencor)

Paul L. submitted info on this based on the OIML Document, also info on what is now being asked by the lab

CONCLUSION: *Cover this at a later time.*

7. Definitions - Software Based Device, etc.

CONCLUSION: *Still need to work on Built for Purpose, Not Built for Purpose and Software based device, See discussion under item 1.*

----

8. Next meeting

At this time, there is consensus of the group to have the next meeting in April. In conjunction with Lab meeting

This will need to have Board Discussion for funding
Meeting will take place in January '07 for the NCWM Board to discuss

It was STRONGLY suggest that all members of the Software Sector review and understand the documents on http://www.welmecwg7.ptb.de/Guides/guides.html

Jim Truex commented that this meeting has been productive. He also indicated that many people are very interested in the work of the Sector and will be watching the progress closely.

Charlene Numrych: Believes that the group needs to document what is currently done with various types of device. This could be taken directly from NCWM Publication 14. These various sections of Pub 14 could then be pulled together into a single document. Need a document of what we do now, agenda item could be to take what we have out of the current Pub 14's and have this pulled together in a single document.

Wayne Stiefel commented that the group needs to focus on the requirements that are found in the OIML and WELMEC documents. If we agree with these requirements, then evaluation can evolve from them.

Mike Cleary commented that the Sector needs to outline the strategic goals. Currently the group tends to jump from on topic to another. An outline would help to guide this work and provide focus for the group.

Respectfully submitted by:

Stephen Patoray

Revised: 12/12/06 (jt)

Meeting Summary for the Second Meeting of the Software Sector
October 18-19, 2006
Annapolis, MD

Repeat of the 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.

<u>There were several general items presented by the Tech Advisor to begin the dialogue.</u>

A) During a time after the previous meeting, there was significant activity on the Software Sector listserv. However, more recently this activity has been minimal. This is a new Sector with many new members and a very difficult task. With all that we have going on it is possible that focus on the work of Software Sector was lost to our regular duties of our "real" jobs. Also, in review of the information that was sent out, it may not have been stated clearly enough what the expectations were for the working groups. To date, I have not received any reports back from the working groups. This is in no way criticism of the groups

SUMMARY: *This led to a discussion regarding previous emails. It was clear that everyone may not have received all of the emails via the listserv. It was also apparent that not all of the current interested parties were on the listserv. The NTEP staff will verify and double check that all interested parties will be on the listserv. Previous emails will be sent to the current members of the listserv.*

B) There has been a suggestion that at the start of the upcoming meeting, the groups be given a short time (30 minutes) to review what they have completed. However, there was concern that many of the people have volunteered for more than one working group. It would be difficult to be in more than one discussion at a time. Also, it would be difficult to have so many conversations going on in one room..

SUMMARY: *There was a brief discussion of each of the action items, but it became obvious that there would need to be more detailed discussion on each of the items.*

C) Another suggestion was for the group to begin the discussion by attempting to answer the question: *Should we make an effort in all that we do to harmonize with OIML, or some other current standard?* It was suggested that with such a decision, it may be more clear which direction each of the Action Items should go. It may also limit the amount of work needed to develop each of the Action Items. On the other hand, it may be the will of the group to not go in that direction due to the content or philosophy of the current OIML (or other) standard and this is not the direction that this group believes is best for evaluation, certification and verification of these types of devices under the US system of W&M.

SUMMARY: *It was clear from the discussion that not all in the room agreed with all parts of the WELMEC, FDA or OIML documents. However, it was clear that it would be useful to use as much from these documents as possible.*

Jim Truex repeated that OIML standards are international standards and we are obligated to look at these various documents and attempt to align wherever possible. He also indicated that all the documents are able to be modified when there is technical justification for such changes.

D) The previous discussion led to a discussion on "first final". Dave Vande Berg wanted to propose that this term be more clearly defined, or the definition be changed to "initial gross quantities".

Jim Truex paraphrased the current definition from NCWM Publication 14 Administrative Policy. Section C. DEVICES TO BE SUBMITTED FOR TYPE EVALUATION TYPE EVALUATION states:

> "In general, type evaluations will be conducted on all equipment that affect the measurement process or the validity or 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."

After a brief discussion, Jim Truex called for the question: Should there be a change to the definition of "First Final"

CONCLUSION:
There was an actual vote taken, with:

1 in favor of changing the definition
27 were not in favor of changing the definition, (leave it as it currently is)

Item 1. Software Identification [model/version, use of help screen, etc.?]

a. Built for Purpose
b. Not Built for Purpose
c. version number or greater

John Roach (CA), Todd Lucas (OH), Paul Lewis (Rice Lake), Bob Hoblit (IBM), Mike McGhee (Actaris), Dave Vande Berg (Vande Berg Scales), Chris Scott (Gilbarco)

SUMMARY: *There was a lengthy discussion on this item/topic. This also carries into several other agenda items. 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 was also the suggestion to use the definitions from the WELMEC document for Type P and Type U instruments. This was modified per suggestions to include the words highlighted in yellow. It was also suggested that a list of examples be provided.

DEFINITIONS:

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 is likely to contain many of the 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 legally relevant functions.

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

Then there was a discussion on specific sections of the WELMEC document that deal with TYPE P and TYPE U requirements. Comments and Question are highlighted in yellow.

P1: Documentation

In addition to the specific documentation required in each of the following requirements, the documentation shall basically include:

a. A description of the legally relevant software.

b. A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).

c. A description of the user interface, menus and dialogues.

d. The unambiguous software identification.

e. An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.

f. The operating manual.

This should not be an issue,

P2: Software identification

The legally relevant software shall be clearly identified. An identification of the software shall be inextricably linked to the software itself. It shall be presented on command or during operation.

We should have addressed this in previous notes. Action item 1

P3: Influence via user interface

Commands entered via the user interface shall not inadmissibly influence the legally relevant software and measurement data.

There shall be a means to prevent changes from the user interface to any metrologically significant portions of the software and measurement data without authorization.(?) May need to define authorization per HB 44.

P4: Influence via communication interface

Commands inputted via communication interfaces of the instrument shall not inadmissibly influence the legally relevant software and measurement data.

There shall be a means to prevent changes from the communication interface to any metrologically significant portions of the software and measurement data without authorization. (?) May need to define authorization per HB 44.

P5: Protection against accidental or unintentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software and measurement data shall be protected against accidental or unintentional changes.

P6: Protection against intentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software shall be secured against the ~~inadmissible~~ unauthorized modification, loading or swapping of hardware memory.

P7: Parameter protection

~~Parameters that fix legally relevant characteristics~~ Metrologically Significant Parameters of the measuring instrument shall be secured against unauthorized modification.

U1: Documentation

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

a. *A description of the legally relevant software functions, meaning of the data, etc.*

b. *A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).*

c. *A description of the user interface, menus and dialogues.*

d. *A legal software identification.*

e. *An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.*

f. *An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.*

g. *The operating manual.*

This one should be acceptable.

U2: Software identification

The ~~legally relevant~~ (metrologically significant, [find and replace]) software shall be clearly identified. An identification of the software shall be inextricably linked to the software itself. It shall be determined and presented on command or during operation.

Inextricably (cannot be separated)
AB: put a note in the checklist for the lab that they cannot "change" the ID?
This should be covered in permanence of marking

U3: Influence via user interfaces

Commands entered via the user interface shall not inadmissibly influence legally relevant software and measurement data.

Use words from P3

U4: Influence via communication interface

Commands or other inputs via ~~non-sealed~~ communication interfaces of the device shall not inadmissibly influence the legally relevant software and measurement data.

There are question on "sealed" this may not be a physical seal.
Being a U, the person selling, may not know about all of the interfaces

There shall be a means to prevent changes from any communication interface to ~~any~~ metrologically significant portions of the software and measurement data without authorization. (?) May need to define authorization per HB 44.

Means to prevent??
The word Commands, may limit what will need to be evaluated. That may be the intent

Done for now.

U5: Protection against accidental or unintentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software and measurement data shall be protected against accidental or unintentional changes.

OK

U6: Protection against intentional changes

~~Legally relevant~~ (metrologically significant, [find and replace]) software and measurement data shall be secured against ~~inadmissible~~ unauthorized modification.

OK

U7: Parameter protection

~~Legally relevant~~ (metrologically significant, [find and replace]) parameters shall be secured against unauthorized modification.

Specifying Notes:

1. Type specific parameters are identical for each specimen of the type and are in general part of the program code i.e. part of the legally relevant software. Therefore requirement U6 applies to them.

2. Device specific parameters:

“Secured” parameters may be changed using an on-board keypad or switches or via interfaces but only *before* the action of securing. Because device specific parameters could be manipulated using simple tools *on universal computers they shall not be stored in standard storages of a universal computer*. Storing of these parameters is acceptable only in additional hardware.

Settable device specific parameters may be changed after securing.

OK

U8: Software authenticity and presentation of results

Means shall be employed to ensure the authenticity of the ~~legally relevant~~ (metrologically significant, [find and replace]) software. The authenticity of the results that are presented shall be guaranteed.

Had discussion on this on Weds,
RM, there is a method to ID that this is the actual software, trace update,

It shall not be possible to fraudulently simulate approved ~~legally (MS) relevant~~ *(metrologically significant, [find and replace])* software using simple software tools.

Definition for simple software tools, e.g. text editor, notepad, office tools, and other commonly available software tools.

U9: Influence of other software

The ~~legally relevant~~ (metrologically significant, [find and replace]) software shall be designed in such a way that other software does not inadmissibly (??) influence it.

This is DOOM!

A suggestion to consider a metrological device table was presented to the group. After modifications were made, the following table was discussed.

General Marking of Metrological Devices

	Software Only (this is U)	Software + Hardware (this is P)	Hardware Only (this is neither P nor U, mechanical)
Make	X	X	X

Model	X	X	X
Revision/Version	X	X	
COC	X	X	X
Serial Number		X	X

CONCLUSION: *It is apparent a lot more study and understanding of these complex issues are necessary. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

Item 2. Software Protection/Security

2a. Identification of unapproved/unauthorized software

Mike Roach (Verifone), Norm Ingram (CA), Bob Hoblit (IBM), Andre Elle (Endress& Hauser), Rich Miller (FMC), Dave Vande Berg (Vande Berg Scale), Chris Scott (Gilbarco), Doug Bliss (Mettler-Toledo)

There is no way to tell someone how to do sealing, you can say what needs to be accomplished.
JT, segregation of parameters is currently allowed. (table of sealable parameters)

JT right now there are two methods, physical seal, audit trail, does the group believe that there needs to be some other category.

NOTES:
Norm was attempting to get input on what current protection is in place, used by manufacturer

Email chain on this item.

There is a discussion on this item.
JT asking the question: will audit trail work for "sealing" software?

Currently, industry does protect, but it is not audit trail.
There is an issue of audit trail, if the software is not running, or have a software service, the changes could be made and not tracked by audit trail.

Checksum:

DB the only way to know is by a bit by bit comparison, which is not practical.
BF uses the checksum in Taxi meters.

?? There is no way to tell someone how to do sealing, you can say what needs to be accomplished.

SW has several examples of methods of sealing.
authentication

Access control
There is also a spec for certification

X509 Certificates,
PCATS certifies vendors
Version Number, application (checksum) There is a challenge response with different certifications. They validate who they are, there may also be limits set. Receive data verification.

JT, segregation of parameters is currently allowed. (table of sealable parameters)

JT right now there are two methods, physical seal, audit trail, does the group believe that there needs to be some other category.

DB: does not believe that HB 44 does not need to be changed.

DO: Needs to know that software is not being manipulated,
SW: What are we attempting to solve?
DO: POS system not certified, made change, got CC, then, there was ability to change version on the screen, WI-WO, Manual Weight entry as a sealable parameter.
SW: X509 Certification, it is something like verisign, electronic signature and verification.
JT could someone put something together?
WS: Canada has a section

DB: Scale System Controller
The scale system controller has approval certifications for USA and the European Union. In this case, a Commercial Off The Shelf (COTS) PC is used in conjunction with a scale system (terminal and weigh platform). The scale system provides the PC with approved gross weight and accepts commands to zero the weight indication. The PC application program

- stores and recalls weights
- computes net weight using a stored weight or manually entered weight
- provides the user display of net weight
- may compute price based on the net weight and a selected commodity code
- may print a weigh ticket

Protection of configuration and price parameters
Metrologically significant parameters are maintained within the scale terminal and are controlled there. Other parameters are stored in a password protected database. The user controls password protection access and distribution.

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

Protection of software
Metrologically significant software is supplied only as binary code. Each such module is protected by a CRC32 checksum. The expected checksums, revision levels, and dates are kept in an encrypted configuration file. If run-time values differ from expected values the system will not operate. The configuration information can be recalled by an inspector using the Help/About menu in the application program.

Protection of active data
Data from the scale terminal is wholly owned by the scale server metrological interface. No other agent can acquire that data when the scale server is running, and the application program will not accept data except from the scale server.
Transactional information is stored in an encrypted Alibi Memory log. No access is permitted to this data except via the supplied application program. Data can be exported via the application program for external use, but no user modifications are permitted to the original transaction data.

Protection of operating system user interface
There are no special restrictions to the operating system. The application program runs as any other on the PC and can be started, stopped, or minimized.

JP: could you replace the word "data" to software or program.
DVB: if the code is complied, there is little chance of changing the code
DO: Facilitation of Fraud has changed since it was first put in the HB. Industry needs to define what is reasonable.
AB: can we take some of the info from WELMEC and use it?

JT: have some agreements but still not in agreement, leave

DB: In Europe, there are things like, safety, highest level security etc. First modification there would be a limit to the risk classes.

P5: Protection against accidental or unintentional changes
Legally relevant 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 of 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.

Validation Guidance: Typical Examples
Checks based on documentation:
□ Check that a checksum of the program code and the relevant parameters is generated and verified automatically.
□ Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.
□ Check that a warning is issued to the user if he is about to delete measurement data files.
Functional checks:
□ Check by practical spot checks that before deleting measurement data a warning is given, if deleting is possible at all.

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 authorisation, e.g. a dialogue statement or window asking for confirmation of deletion.
□ For fault detection see also Extension I.

CONCLUSION: *There is a need for evaluation and work on this area. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

Item 3. Storage of Data [subsections, automatic storing and transmission]

Rich Miller (FMC), Keith Harper (Gencor), Dave Hoffman (Toptech),
Mike Roach (Verifone)

After much discussion the question was raised whether NTEP should dictate how data is stored or require security and allow the manufacturer to design the means, as is done with other devices.

CONCLUSION: *Doug Bliss, motion to place item 3 on low priority and discuss at a later date. Table this indefinitely, This is not outside the realm, but choose not to look at it at this time.*

In Favor: most of the room
Oppose 2

The item will be removed from the agenda at this time but the group will have the opportunity to bring this back up at ay future meeting.

Item 4. Software Maintenance and Reconfiguration

Wayne Stiefel (NIST), Tony Herrin (Cardinal), Gary Lameris (Hobart), Rich Miller (FMC), Bob Hoblit (IBM), George Brazis (Avery), Travis Gibson (Rice Lake), Keith Harper (Gencor)

NOTES:

DB User must agree with the upgrade.
JT: Question, Storage of Data, current NTEP CC, requirements. MSC's
NI: don't we already do this (section 4)
JT does this group need to look at this?
AB: question about if an update, does it need to reconfigure the system?
WS: this should be part of the process since that you are doing the other areas
JT: How many believe that we do not have to address this

Does the sector need to address this issue?

There was a split vote,

CN: need to include the other side of the flow chart to include physical seals.

OIML D-SW 5.2.6.
Only versions of legally relevant software that conform with the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also on the kind of instrument under consideration. The following options 5.2.6.1 and 5.2.6.2 are equivalent alternatives. This issue concerns verification in the field. Refer to chapter 7 for additional constraints.

This follows the traced update, the verified update is still an option

This appears to be covered by Cat 3 and enforcement.
This may appear to be covered by other sections or security
This section should not include eproms
WS: is there a security key?

AB: does it download correctly?
AB: if they are downloading software, then it may need to be Cat 3
WS: OIML says that the audit trail needs to be updated
AB: this may be only guidance in Pub 14 and not a change to HB 44.

The following flow chart, developed to assist the manufacturer/designer was discussed in depth.

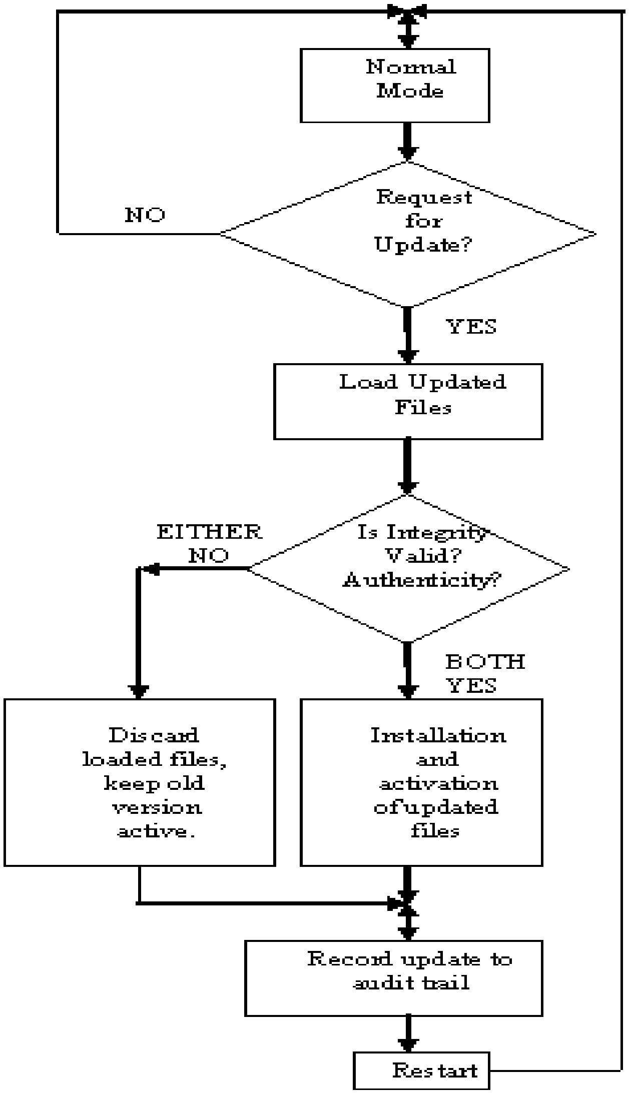

Normal Mode
Request for Update?
NO
YES
Load Updated Files
Is Integrity Valid? Authenticity?
EITHER NO
BOTH YES
Discard loaded files, keep old version active.
Installation and activation of updated files
Record update to audit trail
Restart

Figure 1.0 Traced Update Requirements

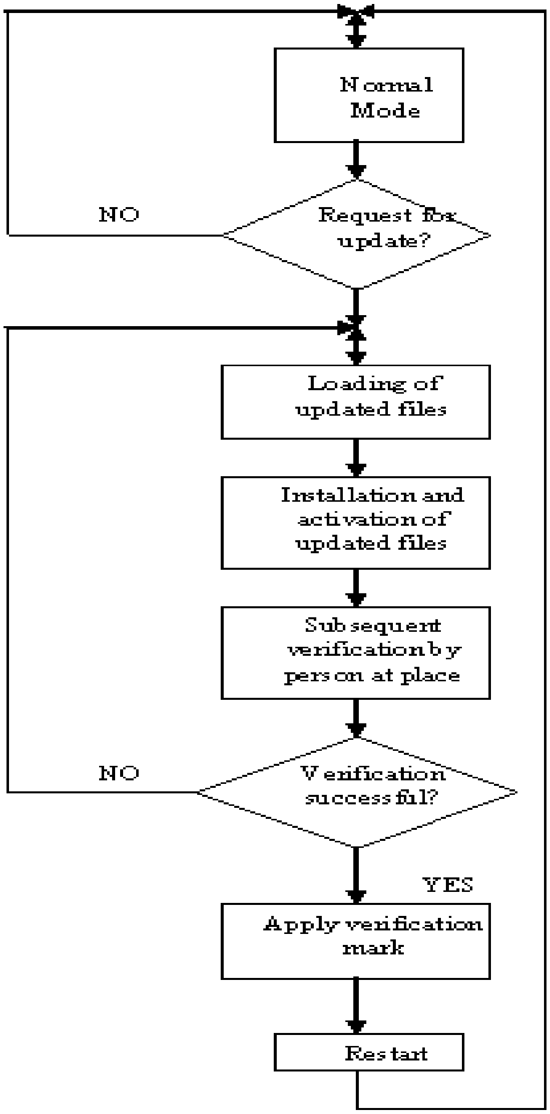


Figure 2.0 Verified Update Model

CONCLUSION: *It is apparent a lot more study and understanding of these complex issues are necessary. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

5. *V*erification in the field, needs work

Jim Truex (OH), Jim Sexton (Rice Lake), Keith Harper (Gencor)

Jim T. Field inspector will need help in the field with verification. Ohio

CONCLUSION: *Cover this at another time. Ohio has developed a field checklist that may be used as a starting point.*

6. NTEP Application – [mfg documentation to be submitted]

Steve Patoray (NTEP), Paul Lewis (Rice Lake), Keith Harper (Gencor)

Paul L. submitted info on this based on the OIML Document, also info on what is now being asked by the lab

CONCLUSION: *Cover this at a later time.*

7. Definitions - Software Based Device, etc.

CONCLUSION: *Still need to work on Built for Purpose, Not Built for Purpose and Software based device, See discussion under item 1.*

----

8. Next meeting

At this time, there is consensus of the group to have the next meeting in April. In conjunction with Lab meeting

This will need to have Board Discussion for funding
Meeting will take place in January '07 for the NCWM Board to discuss

It was STRONGLY suggest that all members of the Software Sector review and understand the documents on http://www.welmecwg7.ptb.de/Guides/guides.html

Jim Truex commented that this meeting has been productive. He also indicated that many people are very interested in the work of the Sector and will be watching the progress closely.

Charlene Numrych: Believes that the group needs to document what is currently done with various types of device. This could be taken directly from NCWM Publication 14. These various sections of Pub 14 could then be pulled together into a single document. Need a document of what we do now, agenda item could be to take what we have out of the current Pub 14's and have this pulled together in a single document.

Wayne Stiefel commented that the group needs to focus on the requirements that are found in the OIML and WELMEC documents. If we agree with these requirements, then evaluation can evolve from them.

Mike Cleary commented that the Sector needs to outline the strategic goals. Currently the group tends to jump from on topic to another. An outline would help to guide this work and provide focus for the group.

Respectfully submitted by:

Stephen Patoray

Revised: 12/12/06 (jt)

National Type Evaluation Technical Committee (NTETC)
Software Sector Meeting
May 7-8, 2007
Sacramento, CA

Agenda Items

Carryover Items		**Page**
1.	Review	
	a) NTETC Software Sector Mission	2
	b) NCWM/NTEP Policies – Issuing CCs for Software	3
2.	Definitions for Software Based Devices	4
3.	Software Identification / Markings	6
4.	Identification of Unapproved / Unauthorized Software	7
5.	Software Protection / Security	10
6.	Software Maintenance and Reconfiguration	10
7.	Verification in the Field, By the Inspector	17
8.	NTEP Application	17
New Items		
9.	S&T Item 310-1 / G-S.2 Facilitation of Fraud	17
10.	Next Meeting	18

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.

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.

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.

Recommendation:

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.

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.

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 legally relevant 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, sites three main areas of : 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 R76 *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.

Conclusion and Recommendation: The above proposed definition was agreed upon to move towards submitting to the S&T. This change would clarify and define what we currently do in NTEP and represent it properly in NIST HB44 to assist the inspector.

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.

Recommendation:

TYPE U Shall meet one of the methods:

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

[1] – Only if no means of displaying this information is available
[2] – Information on how to obtain the remaining items (Make/Model, Version/Revision) shall be included on the C of C.

TYPE P Shall meet one of the methods

Method	NTEP CC No.	Make/Model/Serial No.	Software Version/Revision
Hard-Marked	X	X	Not Acceptable
Continuously Displayed	X	X	X
By command or operator action	Not Acceptable	Not Acceptable	X

Note: Information on how to obtain the remaining items (Make/Model, Version/Revision) shall be included on the C of C.

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.

The sector will forward these items, when completed, to the Regional S&T committees for consideration.

4. Identification of Unapproved/Unauthorized Software

Source: NTETC Software Sector

Background: During the last meeting much discussion was generated. Many comments were addressed.

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

Right now there are two methods, physical seal, audit trail, does the sector believe that there needs to be some other category?

Currently, industry does protect software, but it is not audit trail.
There is an issue of audit trail, if the software is not running, or have a software service, the changes could be made and not tracked by audit trail.

There is no way to tell someone how to do sealing, you can say what needs to be accomplished.

Examples of methods of sealing.
authentication
access control
X509 Certificates,
PCATS certifies vendors

Version Number, application (checksum) There is a challenge response with different certifications. They validate who they are, there may also be limits set. receive data verification

The sector was in general agreement that HB 44 does not need to be changed.

The sector agreed that W&M needs to know that software is not being manipulated,

X509 is a standard for a public key infrastructure (PKI). This is a system where a third party holds the key to decode an encrypted program, to ensure no one messes with it

Scale System Controller
The scale system controller has approval certifications for USA and the European Union. In this case, a Commercial Off The Shelf (COTS) PC is used in conjunction with a scale system (terminal and weigh platform). The scale system provides the PC with approved gross weight and accepts commands to zero the weight indication. The PC application program

- stores and recalls weights
- computes net weight using a stored weight or manually entered weight
- provides the user display of net weight
- may compute price based on the net weight and a selected commodity code
- may print a weigh ticket

Protection of configuration and price parameters
Metrologically significant parameters are maintained within the scale terminal and are controlled there. Other parameters are stored in a password protected database. The user controls password protection access and distribution.

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

Protection of software
Metrologically significant software is supplied only as binary code. Each such module is protected by a CRC32 checksum. The expected checksums, revision levels, and dates are kept in an encrypted configuration file. If run-time values differ from expected values the system will not operate. The configuration information can be recalled by an inspector using the Help/About menu in the application program.

Protection of active data
Data from the scale terminal is wholly owned by the scale server metrological interface. No other agent can acquire that data when the scale server is running, and the application program will not accept data except from the scale server.

Transactional information is stored in an encrypted Alibi Memory log. No access is permitted to this data except via the supplied application program. Data can be exported via the application program for external use, but no user modifications are permitted to the original transaction data.

Protection of operating system user interface
There are no special restrictions to the operating system. The application program runs as any other on the PC and can be started, stopped, or minimized.

In Europe, there are things like, safety, highest level security etc. First modification there would be a limit to the risk classes.

P5: Protection against accidental or unintentional changes
Legally relevant 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 of 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.

Validation Guidance: Typical Examples
Checks based on documentation:
□ Check that a checksum of the program code and the relevant parameters is generated and verified automatically.
□ Check that overwriting of data cannot occur before the end of the data storage period that is foreseen and documented by the manufacturer.
□ Check that a warning is issued to the user if he is about to delete measurement data files.
Functional checks:
□ Check by practical spot checks that before deleting measurement data a warning is given, if deleting is possible at all.

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.

Discussion: At this point around the room there was a great deal of discussion. It was pointed out, that it would be difficult, if not impossible for the NTEP evaluated software to identify if unauthorized software was "added" to the device. It is not possible to identify all unapproved software (e.g. add on software, pirated software).

There was general agreement that this may be a field enforcement issue and that it was not appropriate to continue discussion on this item at this time.

Recommendation: The sector recommended moving this item under agenda item 7, as a sub-item, for discussion at a future meeting.

5. Software Protection / Security

Source: NTETC Software Sector

Background:

Discussion: The discussion from the last meeting on this issue is mingled in item 4. Appropriate sections need to be pulled out by the sector.

The sector reviewed the applicable items, line by line in the MID Software Work Package 2 and the OIML TC9/SC1 R-76-1 Draft Recommendation to determine items appropriate for the evaluation checklist.

Recommendation: Jim Pettinato from FMC Technologies, agreed to pull together additional information regarding the checklists that we have just developed for this section.

6. Software Maintenance and Reconfiguration

Source: NTETC Software Sector

Background: After discussion during the 10/06 meeting, it appeared these issues may go beyond the scope of current NTEP procedures, and possibly NTEP resources. The question was asked, does the sector need to address this issue? There was a split vote, no consensus, so it remains on the agenda.

OIML D-SW 5.2.6. was discussed. Comments included:

Only versions of legally relevant software that conform with the approved type are allowed for use (see OIML D-SW 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation.

It may differ also on the kind of instrument under consideration. The following options OIML D-SW 5.2.6.1 and 5.2.6.2 are equivalent alternatives. This issue concerns verification in the field. Refer to OIML D-SW chapter 7 for additional constraints.

Discussion points and questions:

This appears to be covered by Cat 3 and enforcement.
This may appear to be covered by other sections or security.
This section should not include eproms.
Is there a security key?
Does it download correctly?
OIML says that the audit trail needs to be updated.

The following flow chart, developed to assist the manufacturer/designer was discussed in depth.

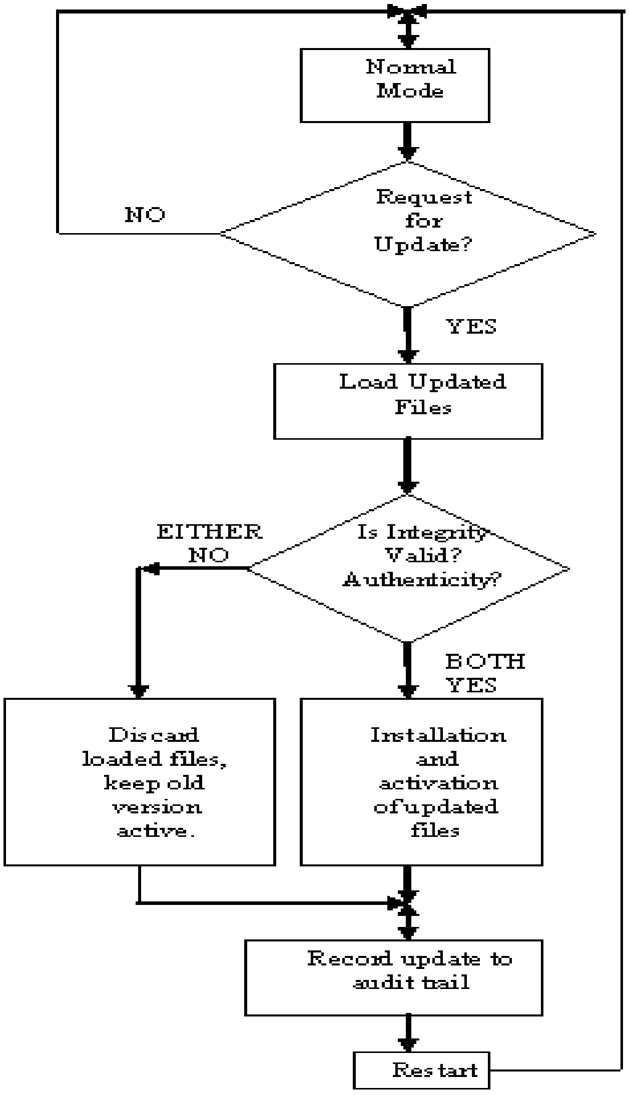

Normal Mode
Request for Update?
NO
YES
Load Updated Files
Is Integrity Valid? Authenticity?
EITHER NO
BOTH YES
Discard loaded files, keep old version active.
Installation and activation of updated files
Record update to audit trail
Restart

Figure 1.0 Traced Update Requirements

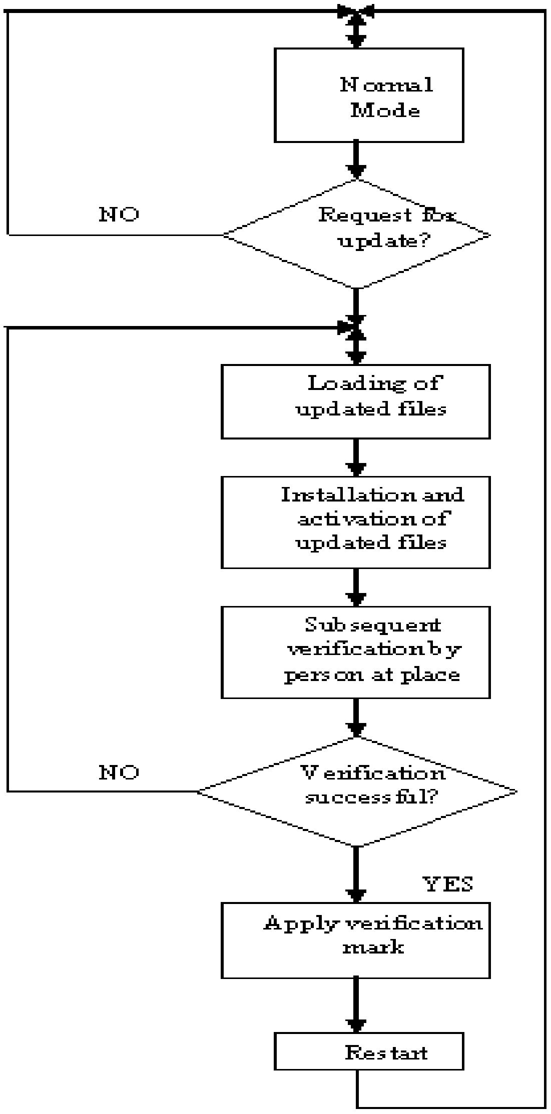


Figure 2.0 Verified Update Model

10/06 Conclusion: *It is apparent a lot more study and understanding of these complex issues are necessary. More discussion will need to take place during the next meeting. Sector members are encouraged to submit specific proposals for consideration.*

Discussion: Traced update provides the ability to update the software either remotely or with equipment that is not part of the device, Category 3 Method of Sealing, it is in line with current technology. It is a feature that currently is being asked for.

Recommendation: After lengthy discussion on this item the sector came to general consensus that the information in the recommendation below should be considered for further developing type evaluation checklists and field test procedures. It was pointed out that these isses are relevant to agenda item 8, the NTEP application.

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.

There was discussion on procedures for verifying the versions of software and it was discussed that these procedures should be part of the NTEP CC.

The sector will continue to develop this area.

This section taken from Document OIML D-SW Working Draft 1 WD

5.2.5 Conformity of production-line devices with the approved type

Requirement: The manufacturer shall produce devices and the legally relevant *(is this term correct?? sap)* software that conform to the approved type and the documentation submitted. There are different levels of conformity demands:

(a) identity of the *legally relevant functions* described in the documentation (6.1) of each device with those of the type (the executable code may differ),

(b) identity of *parts of the legally relevant source code*, and the rest of the legally relevant software complying with (a),

(c) identity of the *whole legally relevant source code*, and

(d) identity of the *whole executable code*.

It has to be defined for each kind of instrument or area of application by the responsible TCs which degree of conformity is suitable. The TCs could define a subset from these conformity degrees for a particular kind of instrument and leave the decision what degree of conformity is to be applied to the approving body.

Except for (d) there may be a software part with no conformity requirements, if it is separated from the legally relevant part according to.5.2.1.2.

Means described in 5.1.1 and 5.2.1 shall be provided to make the conformity evident.

5.2.6. Maintenance and re-configuration

Requirement: Only versions of legally relevant software that conform with the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also on the kind of instrument under consideration. The following options 5.2.6.1 and 5.2.6.2 are equivalent alternatives. This issue concerns verification in the field. Refer to chapter 7 for additional constraints.

5.2.6.1 Verified update

The software to be updated can be loaded locally ie. directly on the 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 legally relevant software of a measuring instrument (exchange with another approved version or re-installation) the measuring instrument is not allowed to be used for legal purposes before a (subsequent) verification of the instrument as described in chapter 7 has been performed and the securing means have been renewed (if not otherwise stated in the relevant OIML Recommendation or in the approval certificate). A person responsible for verification must be at place.

5.2.6.2 Traced update

The software is implemented into the instrument according to the requirements for traced update (**5.2.6.2.1** to **5.2.6.2.6**) if it is in compliance with the relevant OIML Recommendation. 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 ie. directly on the measuring device or remotely via a network. The software update is recorded in an audit trail (see **5.2.6.2.5**). The procedure of a traced update comprises several steps: loading, integrity checking, checking of the origin (authentication), installation, logging and activation.

5.2.6.2.1 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.

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

5.2.6.2.3 Technical means shall be employed to guarantee the authenticity of the loaded software ie. that it originates from the owner of the type approval certificate. This can be accomplished eg. 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>**.

5.2.6.2.4 Technical means shall be employed to guarantee the integrity of the loaded software ie. 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 **or become inoperative.**

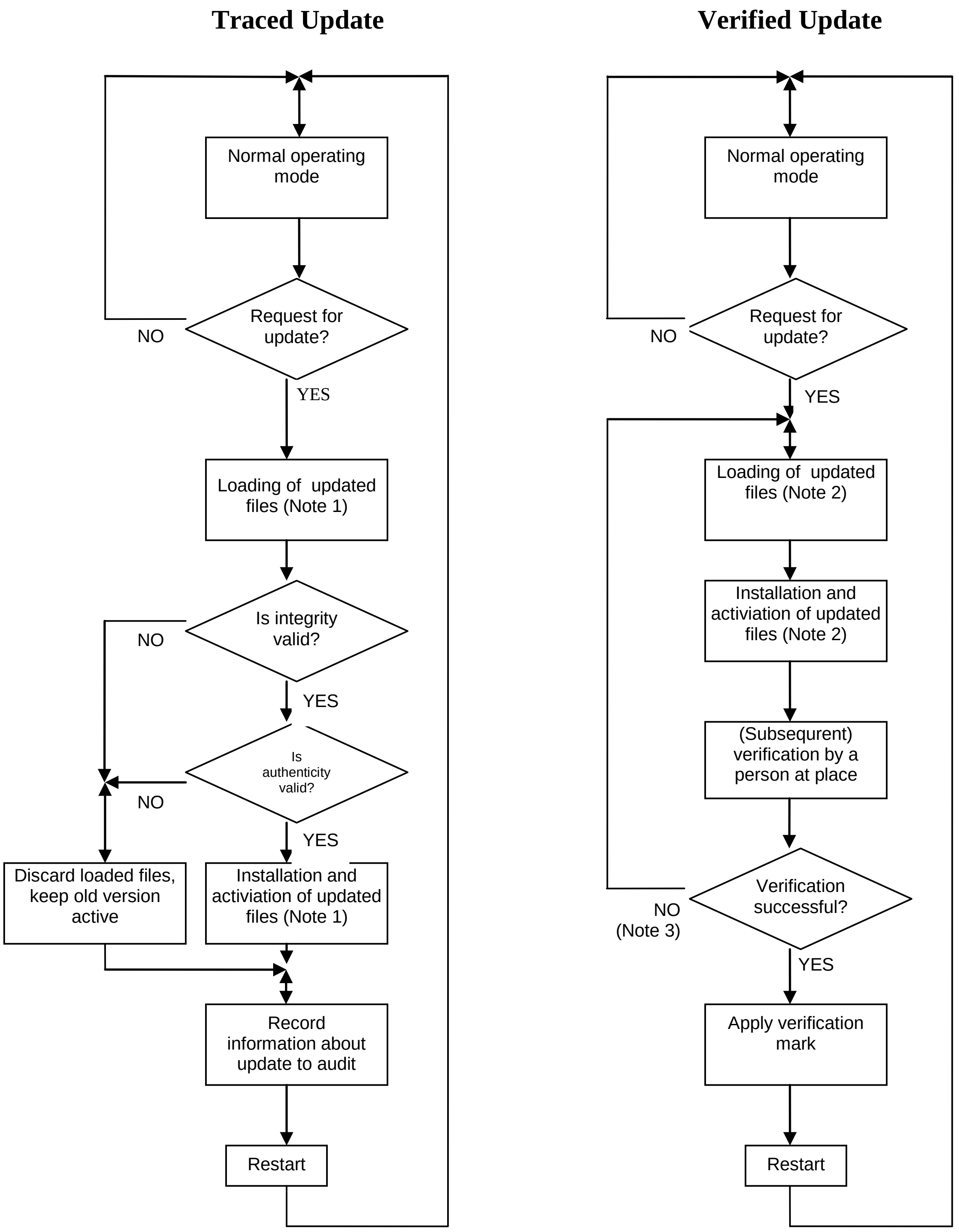
5.2.6.2.5 **The manufacturer shall ensure** ~~**It shall be guaranteed**~~ by appropriate technical means eg. an audit trail that traced updates of legally relevant software are adequately traceable within the instrument for subsequent verification and surveillance or inspection. This requirement enables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

The audit trail shall contain the following information: **notification** ~~**success / miscarriage**~~ of the update procedure, software identification of the installed version, time stamp of the event, identification of the downloading party. An entry is generated for each update ~~**attempt regardless of the success**~~.
The traceability means and records are part of the legally relevant software and should be protected as such. The software used for displaying the audit trail belongs to the fixed legally relevant software. ***Note: This needs to be discussed further due to some manufacturer concerns about where the software that displays the audit trail information is located and who has access if this feature is provided.***

5.2.6.2.6 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. ~~**Relevance of this requirement depends on national legislation.**~~

5.2.6.2.7 If the requirements **5.2.6.2.1** to **5.2.6.2.6** 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 legally relevant and non-relevant software according to 5.2.1.2.
- The whole legally relevant 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

Verified 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 activiation of updated files (Note 1)
Record information about update to audit
Restart
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 2)
Installation and activiation of updated files (Note 2)
(Subsequrent) 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.

7. Verification in the Field, By the Inspector

Source: NTETC Software Sector

Recommendation*:* Cover this at a later time.

8. NTEP Application – [mfg documentation to be submitted]

Source: NTETC Software Sector

Recommendation**:** Cover this at a later time.

NEW ITEMS

9. S&T Item 310-1 / G-S.2 Facilitation of Fraud

Source: NCWM S&T Committee

The S&T Committee has Item 310-1 on their agenda as a voting item. They have requested a position, pro or con, form the NTETC Software Sector. The following is item 310-1 as it appears in NCWM Pub. 16.

Recommendation: Amend Handbook 44, Section 1.10. General Code paragraph G-S.2. as follows:

> **G-S.2. Facilitation of Fraud** - All equipment, and all mechanisms, and devices ~~**attached thereto or used in connection therewith**~~, **without limitation**, shall be so **designed**, constructed, assembled, and installed for use such that they do not facilitate the perpetration of fraud.
> **(Amended 2007)**

The sector reviewed, the Pub 15 and 16 proposals, the Western original,

Background/Discussion: This proposal modifies the language in paragraph G-S.2. to clarify that the prohibition against facilitating fraud applies to the electronically programmed and coded components of weighing and measuring devices to address electronic manipulation or alteration. Some argue the existing language in Section 1.10. General Code. Paragraph G-S.2. Facilitation of Fraud is intended to address only hardware components of weighing and measuring devices. That is, “equipment, mechanisms, and devices” and the mechanics of how they are “constructed, assembled, and installed” appear to deal with tangible components. Fraud issues in the past ten years involved: (1) altering, manipulating, or interfering with software interfaced or installed in equipment; (2) microprocessor issues such as additional pulser units hidden in gas pumps and taximeters; and (3) software programs permitting manipulation of motor truck scale data used to generate weighmaster certificates.

The CWMA, the SWMA, and the WWMA recommended this item move forward for a vote.

The NEWMA recommended this item be referred to the NTETC Software Sector for review and input.

At the 2007 NCWM Interim Meeting, the Committee considered the WWMA proposal and an alternate proposal developed by the SMA. The Committee acknowledged that neither proposal was reviewed by the NTETC Software Sector. The Committee agreed that updating the requirement could be accomplished by adding general terms to address the types of electronic and software-based technology being fraudulently used today. The WWMA proposed language naming specific software applications that should not facilitate fraud. Whereas, the SMA alternate proposal included broader language that is intended to prohibit fraudulent use of software, wireless connections, and all future technology “without limitation.” The Committee agreed that the SMA proposal encompasses all possible equipment configurations and more appropriately addresses the problem at hand. Therefore the Committee agreed to present the SMA proposal for a vote at the 2007 NCWM Annual Meeting.

Discussion: There was lengthy discussion of this item by the sector.

Sector Position: The consensus of the sector, is to support the Central (CWMA) recommendation as a voting item and deleting words “an all” since is was associated with the “attached thereto and …” language. The term “design” adds value. “Software” adds clarification.

> **G-S.2. Facilitation of Fraud** - All equipment, ~~**and all**~~ mechanisms, **software** and devices ~~**attached thereto or used in connection therewith**~~, ~~without limitation~~, shall be so **designed**, constructed, assembled, and installed for use such that they do not facilitate the perpetration of fraud.

10. Next Meeting

The NCWM Board agreed to fund a May 2007 meeting of the NTETC Software Sector. This is the third meeting of the sector in a thirteen month span. The meeting is being scheduled leading into a meeting of NTEP laboratory representatives. The scheduling was intentional, as the decision has been made that it is the “best fit”, in an attempt to have as much NTEP lab(s) representation as possible. Piggybacking meetings also saves travel costs. Therefore, the next planned meeting of the Software Sector will be for the spring of 2008 in adjacent to the NTEP labs meeting.

Discussion: Some members of the sector have expressed concern that waiting a year to meet again may be too long. It was suggested that there always is the possibility of meeting electronically, and taking electronic ballot etc. The other alternative is to ask the BOD to have an additional meeting in the fall.

Conclusion: It was the consensus of the sector to request that another meeting of the Software Sector be held in the fall. This may be in conjunction with one of the Sector Meetings, or as a separate meeting possibly in Ohio. Shortly after the meeting, a request was submitted to the NCWM Board for consideration of funding this meeting.

National Type Evaluation Technical Committee (NTETC)
Software Sector Meeting
October 17 & 18, 2007
Little Rock, AR
DRAFT

Agenda Items

Carryover Items **Page**

New Items

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, sites three main areas of : 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 R76 *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.

NEW DEFITION:

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": exemption from certain requirements (digital load cells and devices with correction methods) this is needed to prevent a "black box" 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 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]

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.

[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.

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 (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 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 itme needs changed to "Identification of Certified Software."

Currently, use version no., ID no., Serial No., however, there is no physical tie to the actual software.

Some international documents, like 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

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 (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)

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

5. Software Protection / Security

The sector spend a significant amount of time reviewing and revamping previous work. OIML and Welmec documents were researched. The following are draft Pub. 14 checklist criteria for consideration at the next meeting.

Building Pub 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
- 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 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, mother board, housing, etc.).

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

TABLE ZZ

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 the 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 fulfil 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 realised 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, eg 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 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 of 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 Pub 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 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 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 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			
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. 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

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 a 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 ie. that it originates from the owner of the type approval certificate. This can be accomplished eg. 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 **or become inoperative**.

Technical means shall be employed to guarantee the integrity of the loaded software ie. 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 **or become inoperative.**

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.

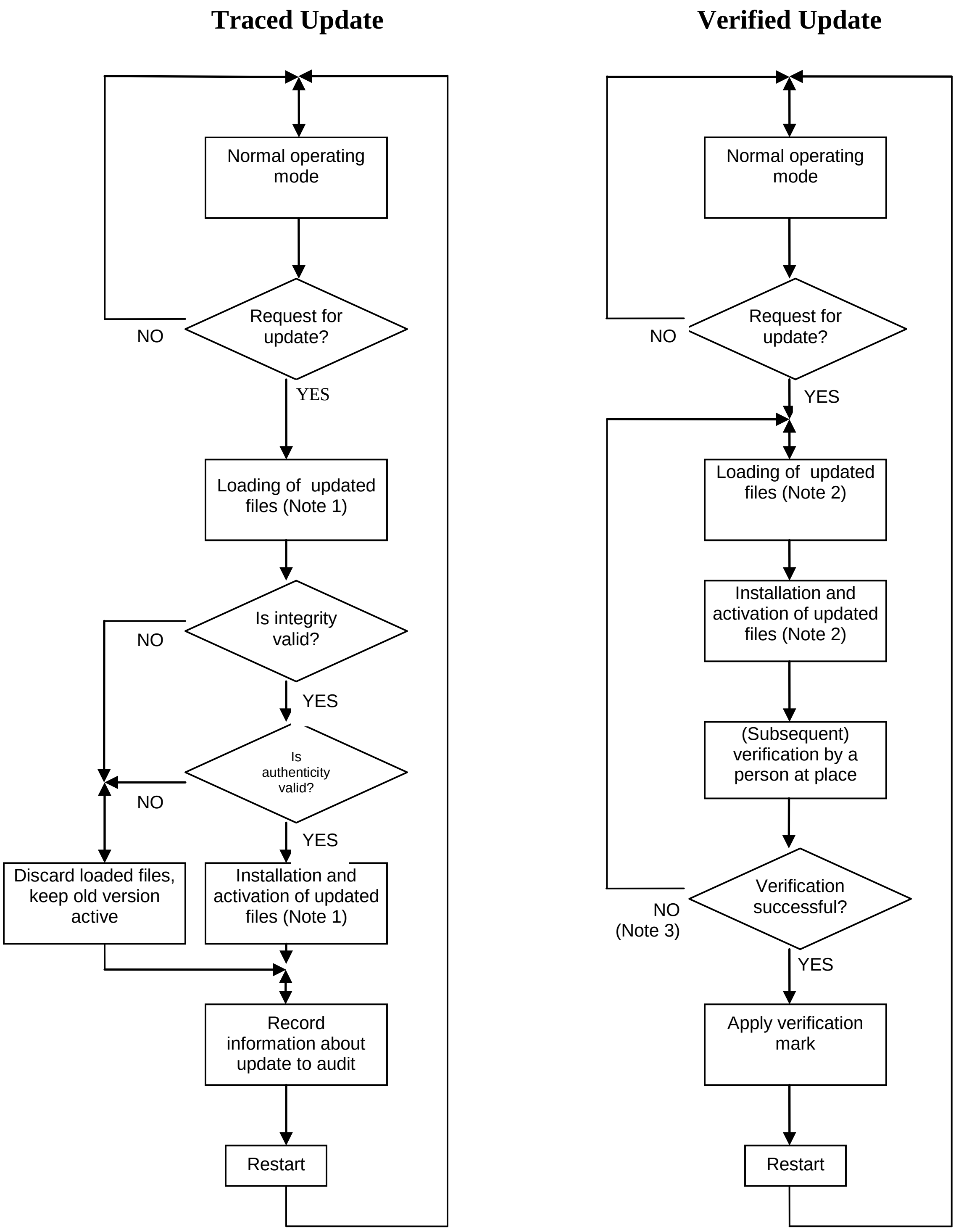

Traced Update
Verified Update
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 1)
Is integrity valid?
NO
YES
Is authenticity valid?
NO
YES
Discard loaded files, keep old version active
Installation and activation of updated files (Note 1)
Record information about update to audit
Restart
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 manufacturer's 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.

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. Sector will be notified as soon as additional information is available.

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.

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.

<u>!!!! Tare data is not included in the metrologically significant software part !!!!</u>

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

<u>Proposed checklist for Pub 14.</u> 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 **<u>Required</u>** 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.

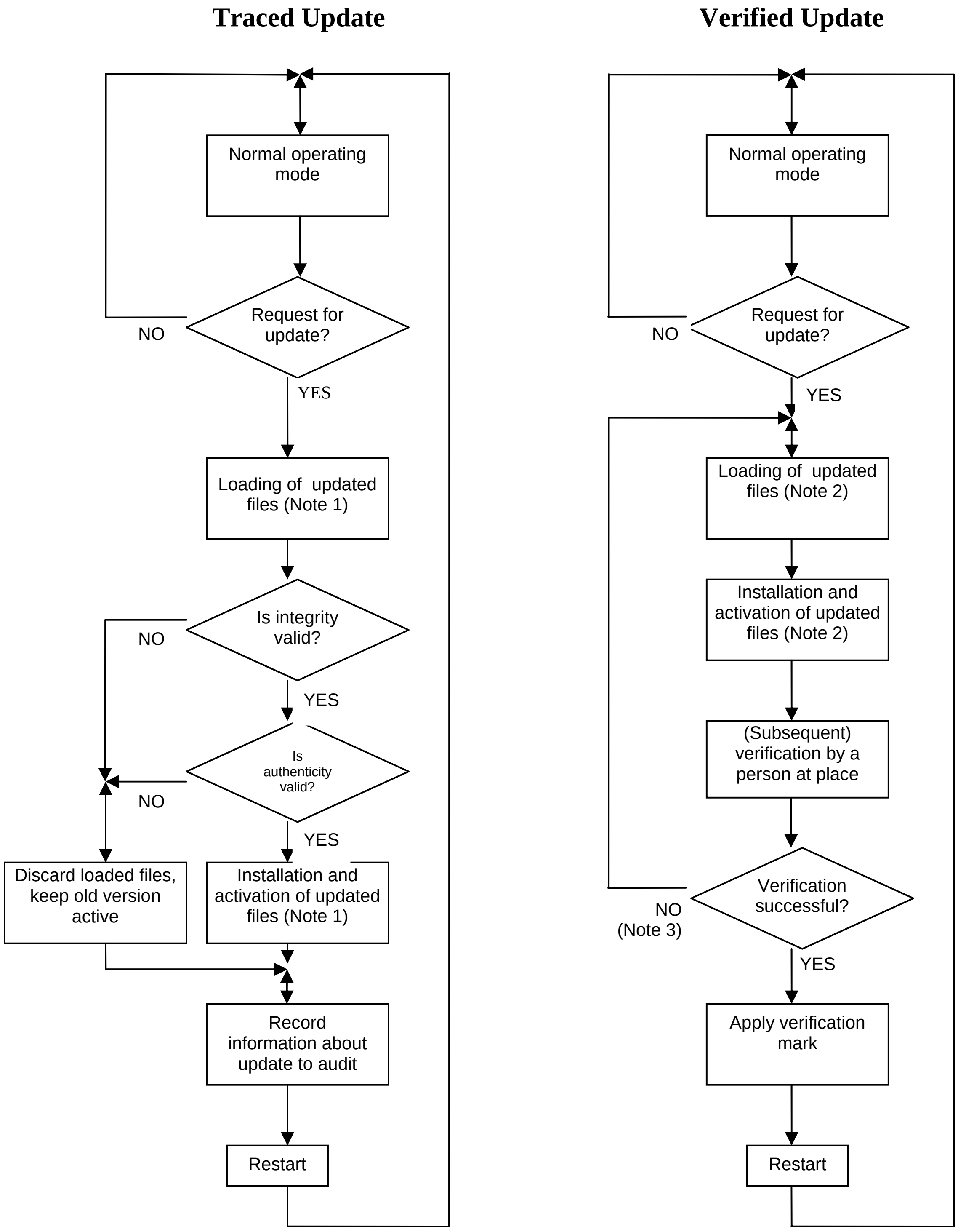

Traced Update
Verified Update
Normal operating mode
Request for update?
NO
YES
Loading of updated files (Note 1)
Is integrity valid?
NO
YES
Is authenticity valid?
NO
YES
Discard loaded files, keep old version active
Installation and activation of updated files (Note 1)
Record information about update to audit
Restart
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 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

Final Summary of Software Sector Meeting May 2008

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.

National Type Evaluation Technical Committee
Software Sector Annual Meeting
March 11-12, 2009 Reynoldsburg, OH

Meeting Summary

Carry-over Items:

National Type Evaluation Technical Committee
Software Sector Meeting
March 11-12, 2009 Reynoldsburg, OH

Meeting Summary

Carry-over Items:

1. Issuing Certificates of Conformance 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. During the 1998 NCWM Annual Meeting, 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."

Also relevant, from Section C of NCWM Publication 14: "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."

Recommendation: The Sector recommended the following language to be submitted to the NTEP Committee as a policy change, and requested that 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 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 NTEP committee included this item in their agenda (*NTEP Committee 2009 Interim Agenda Item 8*). There was no discussion during the open hearing, and it was determined that this item be given voting status for the 2009 Annual Meeting Agenda.

Discussion: Ambler Thompson observed that in reality this type of software represents only a small portion of type evaluations; the vast majority of them are not standalone software. Cassie Eigenmann indicated that this item as written might not clearly state the intention; which is to simply allow the labs to call standalone software packages that are type approved to be categorized as 'software'. It is an administrative change, not really a regulatory change. The labs will not be doing anything differently at type approval time.

Dennis Beattie made the statement that if you follow the concept of 'first final', then you have to address every step of the process (and if that is done with software, then the requirement to address software is obvious). David Vande Berg explained that it is not always black/white, i.e. external software for tare/net calculations is sometimes not judged subject to type approval. It was suggested by Norm Ingram that we should clearly define what we mean by 'software requiring a separate CC'; Cassie Eigenmann recommended using specific examples.

Steve Patoray listed some goals he felt it was important the Sector accomplish:

- Answer the question "What is this item that is up for vote going to change in practice?"
- Address Scale Manufacturers Association's concerns on the S&T agenda items (310-2 and 310-3).

Ambler Thompson agreed, further suggesting that the Sector needs to 'sell' the concepts we've realized; and it was mentioned that the Regional meetings might be an opportunity to approach the states.

Jim Truex (NTEP Administrator) felt that the upcoming vote will be a technical vote, requiring at least 27 states to vote in the affirmative to pass. He also indicated that this will not change the way the labs operate – it is merely the ability for the labs to label evaluated standalone software as such, and not be forced to categorize it as some type of device such as 'weigh-in-weigh-out-system'. Steve Patoray also suggested that this is an important vote for the Sector; and asked that if the states continue to avoid dealing with software what is the future of the Sector?

Conclusions:

- **The Sector feels that this item is important and that there exists the possibility of misinterpretation of the scope/intent of this item by other interested parties, hence the Sector agreed to the following actions:**

o Generate Problem Statement and specify benefits addressed by change	(Done)
o Feedback from labs/inspectors	(Lucas, Frailer, Ingram?)
o 'Sales flyer' / Newsletter article	(Bliss et al)
o Request added as Agenda item at CWMA/NEWMA?	(Pettinato/Ingram)
o Attend CWMA/NEWMA regional meetings?	(?)

NCWM was contacted and the staff indicated that if it is desired to include an article in the newsletter, a final draft must be submitted by April 15th. The Sector work group should have a draft circulating by 4/3/09 so comments can be gathered by 4/10/09 for consideration prior to the final draft.

Doug Bliss provided a draft 'slide show' format presentation as a starting point for clearly presenting the ideas put forth by the Sector, and started on a draft article for the newsletter. Further work has progressed since the meeting *(see Appendices B & C).*

2. Definitions for Software Based Devices (2009 Interim Agenda Item 310-2)

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)

The Sector recommended the following definitions be submitted to the S&T Committee as an item and be considered for inclusion in Appendix D of NIST Handbook 44 to replace the current definition of 'build-for-purpose device':

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.

At the 2009 NCWM Interim Meeting, the Committee received comments from the Scale Manufacturers Association stating that it now opposes this item since there is no technological justification for making a distinction in software-based device types. Darrell Flocken added that the SMA can only provide limited responses; SMA continues to support the efforts of the Software Sector and the SMA response is based on the concern that the proposed definitions in this recommendation and the marking requirements proposed in agenda item 310-3 will require weighing devices be more complex than those currently produced.

The Meter Manufacturers Association indicated that it supports the item as written in the recommendation.

Will Wotthlie, Maryland, did not agree with the SMA position that there are no technological difference between the types of software-based devices. He added that Type P devices and separable elements have limited flexibility in changing software and indications and frequently include the sensing elements necessary for the measurement (e.g., load cells, meters, etc.). Whereas Type U devices and separable elements are typically devices that do not contain measuring elements; can be replaced with compatible equipment and display devices purchased from any number of sources; and only process metrological information received from measuring and other sensing elements.

Stephen Patoray, Consultants in Certification, agrees with the SMA that there are few differences between Type P and U software-based devices. However there are significant differences between Type P and U devices in that a Type P device is defined as an instrument that requires a security means since the instrument has fixed hardware (including sensing components) where the metrological software is ***embedded*** into the instrument. Type U devices do not include fixed components and metrological software can not be sealed using physical security seals or the minimum form of an audit trail (i.e., two event counters).

Software Sector Co-Chair Jim Pettinato (FMC Technologies) added that international recommendations recognize the differences between embedded software and programmable/loadable software. Additionally, the Software Sector recommends that this item remain informational to allow conference members to further study that proposed definitions.

The S&T Committee agreed with the comments received during the open hearing and the request from the co-chairman of the software sector and agreed that this item should remain an Informational item for further review.

Additional background information on this item can be reviewed in the 2009 Interim Agenda (NCWM Pub. 15).

Discussion: It was reiterated by several individuals that again it seems that resistance to this item stems not from a disagreement with the intention, but from either a misunderstanding of the applicability or unrelated concerns over marking requirements.

Further discussion was related to how to best present the opinion/goals of the Sector to the interested external parties, such as the NCWM standing committees and the individual states.
Some discussion on the wording of the definitions took place as well, with the slightly modified version being proposed:

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

(a) Type 'P' (aka built-for-purpose) software-based electronic devices. 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; or

(b) Type 'U' (aka not-built-for-purpose) software-based electronic devices. All metrological software-based devices not meeting the conditions of a Type 'P' device. Example: a personal computer or other device and/or element with PC components with programmable or loadable metrological software.

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

Conclusion: No consensus was reached on any language change. The Sector did agree that including the reason(s) for proposing these definitions as part of the effort to educate/promote external parties would be beneficial; and that we would attempt to explain the reasoning/intent of the proposed definitions together with/as part of the action items for Item 1.

3. Marking of Software Identification – G-S.1 (2009 Interim Agenda item 310-3)

Source: 2008 Carryover Item

Background: Starting at the October 2007 meeting, the Software Sector has discussed the value and merits of required markings for software. After several iterations, the Sector developed a table to reflect their positions:

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.			

This table was submitted to NCWM S&T Committee and was assigned Developing status in 2008.

Prior to the 2009 Interim NIST Weights/Measures Division commented on this item, and presented an alternate proposal with significant modifications, which were included in the Interim Meeting Agenda background for the item (See 2009 Pub 15 for more details).

This item was assigned Informational status for the NCWM 2009 Annual Meeting.

Discussion: It was noted by several Sector members that the perceived scope of the original proposal has been extended by the modifications made by WMD and now appears to exceed both the purview and the intent of the Sector and it has become difficult to discern what our intentions were. Based on the fact that the table seems to have actually made the Sector's intent less clear, it was proposed by the chair to revisit this item in relation to the current text of G-S.1 to clarify exactly what real changes to Handbook 44 would be required to achieve the intent of the Sector. It was also noted that there was some validity to the Scale Manufacturers Association argument that there is no justification for differentiation of marking requirements based on device type (P or U). After additional lengthy discussions, the following modified versions of G-S.1/G-S.1.1 were drafted:

> **G-S.1. Identification. –** All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect **and manufactured** ~~prior to~~**after** **January 1, 201X**, shall be clearly and permanently marked for the purposes of identification with the following information:
>
> (a) the name, initials, or trademark of the manufacturer or distributor;
>
> (b) a model identifier that positively identifies the pattern or design of the device;
>
> > *(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.*
> > *[Nonretroactive as of January 1, 2003]*
> >
> > (Added 2000) (Amended 2001)
>
> *(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts and* ~~***not-built-for-purpose software-based***~~ ***software that is not part of a Type P (built-for-purpose) device.****;*
> *[Nonretroactive as of January 1, 1968]*
>
> (Amended 2003 **and 201X**)
>
> > *(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.*
> > *[Nonretroactive as of January 1, 1986]*
> >
> > *(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).*
> > *[Nonretroactive as of January 1, 2001]*
>
> *(d) the current software version or revision identifier for*~~***-not-built-for-purpose***~~ ***software-based electronic*** *devices;*
> *[Nonretroactive as of January 1, 2004]*
>
> (Added 2003) **(Amended 201X)**
>
> > *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> > *[Nonretroactive as of January 1, 2007]*
> >
> > *(Added 2006)*
> >
> > *(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).*
> > *[Nonretroactive as of January 1, 2007]*
> >
> > *(Added 2006)*

(e) an NTEP Certificate of Conformance (CC) number or a corresponding CC Addendum Number for devices that have a CC. The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003~~, and~~**,** 2006 **and 201X**)

*G-S.1.1. ~~Location~~ Method of Marking Information for ~~Not-Built-For-Purpose~~ all Software-Based Devices. – **For** ~~not-built-for-purpose, software-based~~ **devices** manufactured ~~prior to~~after January 1, 201X, **either**:*

*(a) The required information in G-S.1. Identification. ~~**(a), (b), (d), and (e)**~~ shall be permanently marked or continuously displayed on the device; or*

(b) The Certificate of Conformance (CC) Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."

***Note:** For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*
[Nonretroactive as of January 1, 2004]

(Added 2003) (Amended 2006 **and 201X**)

It was noted that though currently it is allowable to display the CC number via a menu, there has been some challenges locating this information in the field due to the vagueness of the term 'easily recognized'. Hence, since it is left to the interpretation of the NTEP laboratory to ascertain whether a device's method for displaying the CC number meets the requirements, this vagueness has not been addressed in this new recommendation.

John Roach (CA NTEP Lab) indicated that if the proposed table (or some version thereof) is not eventually included as part of G-S.1 that it may be useful to incorporate a suitable table into Pub 14.

Conclusion: The Sector wishes to address concerns related specifically to software and does not wish to debate the merits of general marking requirements beyond that related to software identification. We feel the above proposed changes better reflect the Sector position. If WMD and NCWM S&T feel a table outlining general marking requirements would clarify the intent of G-S.1 then the Sector suggests that following simplified version may better suit the purpose.

Table G-S.1.~~a~~ Identification
for Devices Manufactured on or after January 1, 201X

Required Marking	Full Mechanical Devices and Separable Mechanical Elements	Electronic Devices, Software Based
Manufacturer or CC holder ID	Hard Marked	Hard Marked, Continuously Displayed, or Via Menu (display) or by command or operator action
Model identification	Hard Marked	Hard Marked, Continuously Displayed, or Via Menu (display) or by command (operator action)
Serial number	Hard Marked	Hard Marked, Continuously Displayed [1]
Metrologically Significant Software version	Not Applicable	Continuously Displayed, Via Menu (display) or by command (operator action) [2]
Certificate of Conformance number	Hard Marked	Hard Marked or Continuously Displayed, or Via Menu (display) or by command (operator action)[3]

[1] Type 'U' devices need not have a non-repetitive serial number.

[2] If the manufacturer declares that the primary sensing element "software" is integral, has no end user interface and no print capability, 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).

[3] If the Certificate of Conformance number is to be displayed via menu and/or submenu, the means of access must be easily recognizable. In addition, instructions on how to obtain the remaining required information not hard-marked or continuously displayed shall be included on the NTEP CC.

(Added 201X)

Note that this new version of the table reflects the aforementioned changes proposed for the G-S.1 text as well, homogenizing Type P and Type U requirements, with the exception of the serial number requirement being waived for standalone software. It was also noted that much of the information previously included in the separate proposed Table G-S.1b was redundant as it is already stated verbatim in the text of G-S.1; hence the Sector questions the benefit of the WMD - proposed separate Table G-S.1b.

4. Identification of Certified Software

Source: NTETC Software Sector

Background: This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?" In previous meetings it was shown that the international community has addressed this issue (both WELMEC and OIML). From WELMEC:

> **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**.**

From OIML:

Example from DSW 2CD (now D-31):

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the W&M inspector information that something has changed? (Yes, the Category III audit trail or other means of sealing). How can the W&M inspector identify an NTEP Certified version? (They can't, without adding additional requirements like what is described here, in conjunction with including the identifier on the CoC).

Recommendation: The Sector believes that we should work towards language that would include a requirement similar to the OIML requirement in HB44. It is also the opinion of the Sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

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

> 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)

Initial draft proposed language: (G-S.1.1?)

Identification of Certified Software:

Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified. The identification of the software shall be inextricably linked to the software itself.

- **Unique identifier must be displayable/printable on command or during operation, etc. (marking req't in addition)**
- **At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc (crc32, for example)**

Discussion: Discussion on this item was brief; it was the general consensus that those in attendance understood the goals of this item and were in agreement of those goals; however the conceptual language was not far enough along to warrant detailed discussion specific to a draft proposal and more work offline should be done.

Conclusion: A work group will be designated by the Sector Co-Chairs prior to the NCWM Annual Meeting to further promote the state of this item, to be discussed at the next Sector meeting.

5. Software Protection/Security

Source: NTETC Software Sector

Background: The sector agreed that Handbook 44 already has audit trail and physical seal, but the question on the table is does the Handbook need to be enhanced to sufficiently discourage the facilitation of fraud, intentional or accidental, where software is concerned?

WELMEC and OIML again have addressed this issue specifically when dealing with software. From WELMEC:

> **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 of 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.

Recommendation: The Sector derived a suitable checklist for Pub 14 from the OIML checklist, and asked the current NTEP labs to begin using this checklist on a trial basis for new type approval applications.

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** □

Discussion: The Chair requested feedback from the NTEP Labs as to whether they had the opportunity to utilize the checklist; each lab reported either they have not had any applications for devices where the checklist could be used, or were unaware of the request to try the checklist. The labs were again asked to try to use the checklist should the opportunity present itself.

Conclusion: The Sector will again wait for laboratory feedback on this item; discussion on this item will continue as part of the next agenda item since the two are closely related.

6. Software Maintenance and Reconfiguration

Source: NTETC Software Sector

Background: The Following Items were reviewed by the Sector in previous meetings.

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 <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 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 <u>or become inoperative.</u>
>
> 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 <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

d. 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).* The statement in italics will need to be reworded to comply with US W&M requirements.

Recommendation: The Sector **agreed** that the two definitions below for Verified update and Traced update were acceptable.

> **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.

The Sector also worked towards language proposed for defining the requirements for a Traced Update (currently considered as relevant for Pub 14):

> <u>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.</u>

Use of a Category 3 audit trail is acceptable for the software update logger. In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement.A software update log entry shall include the following:

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

A Category III device may include the software update events in the Category III audit log in lieu of a separate software update log; the existing requirement for 1,000 entries supersedes the requirement for 10 entries.
The traceability means and records are part of the metrologically significant software and should be protected as such. If software separation is employed, 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. Manufacturers did indicate that there are methods available to encrypt the audit trail information; however, it cannot be protected from being deleted.)* (include flowchart from OIML D-31)

Discussion: The Sector discussed how to best move this item forward, and there was also some discussion as to whether new language for the General Code was required. The following new text was proposed:

G-S.9. Metrologically Significant Software Updates

The updating of metrologically significant software shall be considered a sealable event.

Metrologically significant software that does not conform to the approved type is not allowed for use.

Jim Truex indicated that the current requirements in G-S.8 already make the statement that any changes that affect metrological function are sealable, hence software updates may be covered and the proposed G-S.9 unnecessary. Todd Lucas suggested we go ahead and submit the proposed G-S.9 to the Committee and request a clarification/interpretation of G-S.8

Conclusion: The Sector feels that the explicit language proposed for G-S.9 is clearer than any implied requirement in G-S.8. The Sector would like a clarification/interpretation of G-S.8 as it relates to software updates from the S&T Committee (with their response preferably to be included in Pub 16). The Sector will also continue to develop the proposed text (and flow chart) targeted for inclusion in Pub 14.

(Note to S & T – this item assumes additional requirements in individual codes will be eventually added to address this requirement; e.g. L.M.D. code has philosophy of sealing section that could be enhanced to include processes described.)

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

Source: NTETC Software Sector

Background: What tools does the field inspector need as relates to software-based electronic devices? Some possible answers:

NTEP CC – hard marked, continuously displayed, via menu command or operator action
Clear and simple instructions on NTEP CC to get to the other Inspection Information
The metrologically significant software identifier needs to be easily accessible from operator console

Clear and simple instructions on NTEP CC to access audit trail(s)

Recommendation: The Sector needs to continue to develop this item.

Discussion: Some discussion about system information requirements for the inspector took place… does the inspector really need to have access to OS, RAM information, etc? (General opinion seems to be if there is a dependency, then the NTEP lab would specifically include that requirement in the CoC.)

Audit trail info – the question was asked, does there need to be a specific requirement for providing access to this information?

Regarding the concept of First Final – There was some concern expressed as to how the inspectors are able to discern where the indication of first final be found for the SYSTEM (as opposed to the DEVICES in the system). What devices in the system are of concern to the inspector? The NTEP Administrator indicated that field inspectors need to follow the system all the way to receipt/bill generation.

Data transmission is an issue when considering systems as opposed to devices… how far does the inspector's jurisdiction extend? (Should we model future requirements on the WELMEC section concerning DTD/DSD?) Data transmission/storage is not currently being addressed by the Sector at this time.

Since part of the Sector's mission is education, do we want to assist in developing training aids for labs/inspectors related to evaluating/inspecting software-based devices? This will be a topic to be added to the Sector's agenda for the next meeting.

Conclusion: The Sector will continue to develop this item, and initiate a new agenda item specific to inspector training in relation to evaluating/validating software-based devices.

8. NTEP Application for software requiring a separate CC

Source: NTETC Software Sector

Background: This item had been on the agenda of previous meetings, but was not discussed due to time limitations.

Recommendation: Identify issues, requirements and processes for type approving type U device applications.

Discussion: It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. Question: what gets submitted? What requirements/mechanisms for submission should be available?

Validation in the lab - all required subsystems shall be included to be able to simulate the system as installed.

It was noted this agenda item is irrelevant if the NTEP Committee does not approve the pending item up for vote.

John Roach (CA NTEP Lab) stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale labs and scale manufacturers indicated that this is not usually done for scale evaluations.

Conclusion: The Sector will continue to develop this item, contingent on the status of the related NTEP Committee agenda item after the 2009 Annual meeting.

New Items:

9. Sealing requirements for Electronic Devices

Source: Weighing Sector Tech Advisor

Background: Steve Cook of NIST has been involved in attempting to address some concerns with the current wording of G-S.8 as it relates to the sealing of electronic devices and configuration modes. Since this is related in some respects to other items within the purview of the Software Sector, it was suggested that it may be beneficial for the Sector to review and comment on the proposed language.

Discussion: The Sector discussed the relevance of this item, and though it is related somewhat to our discussions on software security and maintenance/reconfiguration, it is broader in scope and hence it was decided that the item was not wholly relevant to the Sector's mission.

Conclusion: The Software Sector takes no position on these proposed changes.

10. Next Meeting

Recommendation: The Sector was asked to develop a proposed date and location for the next meeting.

Discussion: The Sector discussed two options for the next meeting; continuing to meet in Ohio or alternating to a Western location to maintain equity in travel for the various participating labs. There appeared to be a preference (after an informal polling) to alternate the meeting location from year to year.

Conclusion: The Sector recommends that the next meeting be held in Sacramento in or around March 2010. Sector Co-Chair Norm Ingram will investigate suitable hotels and meeting facilities and report back to NCWM. Details need to be firmed up by December of this year.

Appendix A – 2009 Software Sector Attendees

2009 Software Sector Meeting Attendee List
March 11-12, 2009 / Reynoldsburg, Ohio

□ **William Arce**
USDA/Packers & Stockyards
210 Walnut Street, Suite 317
Des Moines, IA 50309
P. (515)323-2510 **F.** (515)323-2590
E. william.arce-arana@usda.gov

□ **Dennis Beattie**
Measurement Canada
400 St Mary Ave
Winnipeg, Manitoba R3C 4K5
P. (204)983-8910 **F.** (204)983-5511
E. dennis.beattie@ic.gc.ca

□ **Doug Bliss**
Mettler-Toledo
1150 Dearborn Drive
Worthington, OH 43085
P. (614)438-4307 **F.** (614)438-4355
E. doug.bliss@mt.com

□ **Cathy Brenner**
USDA GIPSA FGIS
10383 North Ambassador Drive
Kansas City, MO 64153-1394
P. (816)891-0486 **F.** (816)891-8070
E. cathleen.a.brenner@usda.gov

□ **Cassie Eigenmann**
DICKEY-john Corporation
5200 Dickey-john Road
Auburn, IL 62615
P. (217)438-2294 **F.** (217)438-2635
E. ceigenmann@dickey-john.com

□ **Darrell Flocken**
Mettler-Toledo, Inc.
1150 Dearborn Drive
Worthington, OH 43085
P. (614)438-4393 **F.** (614)438-4355
E. darrell.flocken@mt.com

□ **Michael Frailer**
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
P. (410)841-5790 **F.** (410)841-2765
E. fraileml@mda.state.md.us

□ **Travis Gibson**
Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868
P. (715)234-3494 **F.** (715)234-6967
E. tragib@rlws.com

□ **Bob Helwig**
Vande Berg Scales
770 7th Street NW
Sioux Center, IA 51250
P. (712)722-1181 **F.** (712)722-0900
E. bhelwig@vbssys.com

□ **Norman Ingram**
California Division of Measurement Standards
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916)229-3016 **F.** (916)229-3015
E. ningram@cdfa.ca.gov

□ **Paul Lewis**
Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868
P. (715)234-9171 **F.** (715)234-6967
E. plewis@ricelake.com

□ **Todd Lucas**
Ohio Department of Agriculture
8995 East Main Street
Reynoldsburg, OH 43068
P. (614)728-6290 **F.** (614)728-6424
E. lucas@agri.ohio.gov

□ **Ken Mahaney**
Vande Berg Scales
770 7th Street NW
Sioux Center, IA 51250
P. (712)722-1181 **F.** (712)722-0900
E. KMahaney@vbssys.com

□ **Steve Patoray**
Consultants on Certification, LLC
1239 Carolina Drive
Tryon, NC 28782
P. (828)859-6178 **F.** (828)859-6180
E. steve@consultoncert.com

□ **James Pettinato**
FMC Technologies Measurement Solutions, Inc.
1602 Wagner Avenue
Erie, PA 16510
P. (814)898-5250 **F.** (814)899-3414
E. jim.pettinato@fmcti.com

□ **John Roach**
California Division of Measurement Standards
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916)229-3014 **F.** (916)229-3015
E. jroach@cdfa.ca.gov

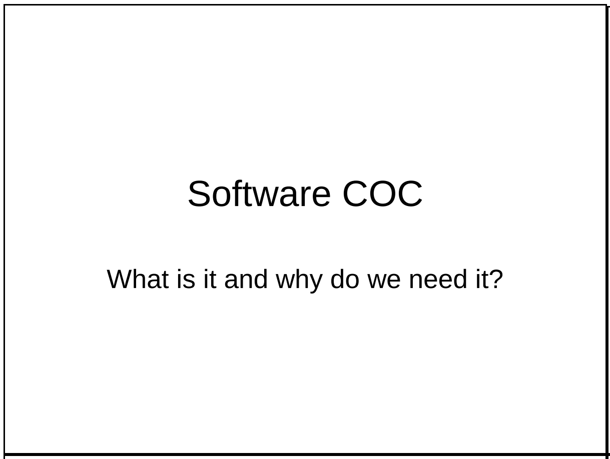
□ **Mark Schwartz**
Accu-Sort Systems, Inc.
511 School House Road
Telford, PA 18969
P. (215)721-5053 **F.** (215)721-5557
E. mark.schwartz@accusort.com

□ **Ambler Thompson**
NIST, Weights & Measures Division
100 Bureau Drive
Gaithersburg, MD 21701
P. (301)975-2333 **F.** (301)975-8091
E. ambler@nist.gov

□ **James Truex**
National Conference on Weights & Measures,
Inc.
88 Carryback Drive
Pataskala, OH 43062
P. (740)919-4350 **F.** (740)919-4348
E. jim.truex@ncwm.net

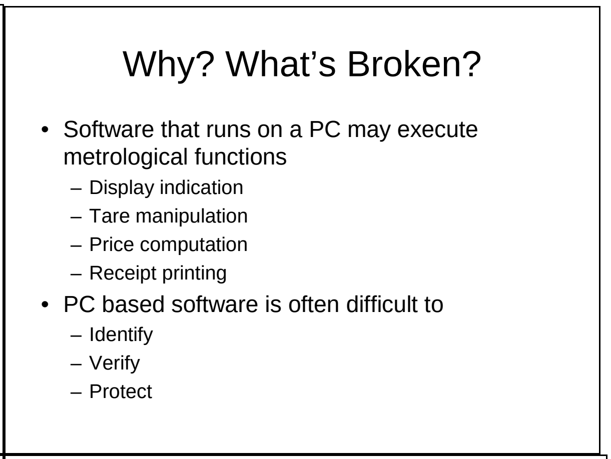
□ **David Vande Berg**
Vande Berg Scales
770 7th St NW
Sioux Center, IA 51250-1918
P. (712)722-1181 **F.** (712)722-0900
E. davevb@vbssys.com

Appendix B – Slide Show (Draft) for Presentation at Regional Meetings



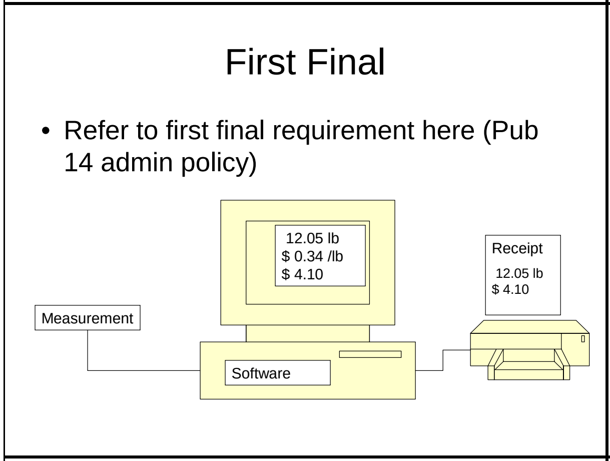
Software COC

What is it and why do we need it?



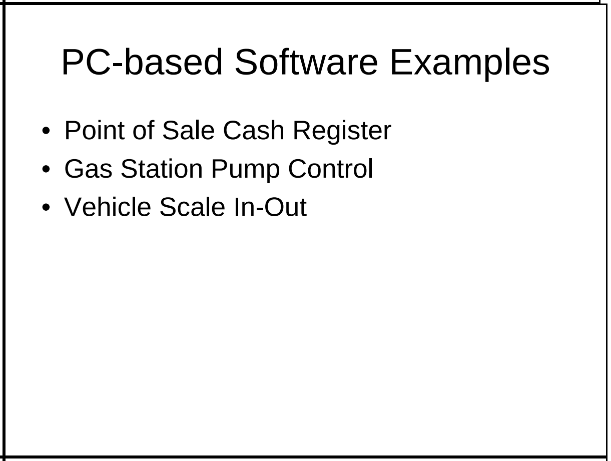
Why? What's Broken?

- Software that runs on a PC may execute metrological functions
  - Display indication
  - Tare manipulation
  - Price computation
  - Receipt printing
- PC based software is often difficult to
  - Identify
  - Verify
  - Protect



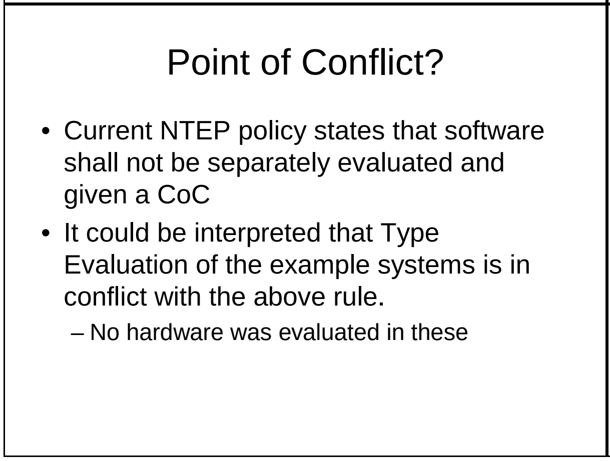
First Final

- Refer to first final requirement here (Pub 14 admin policy)



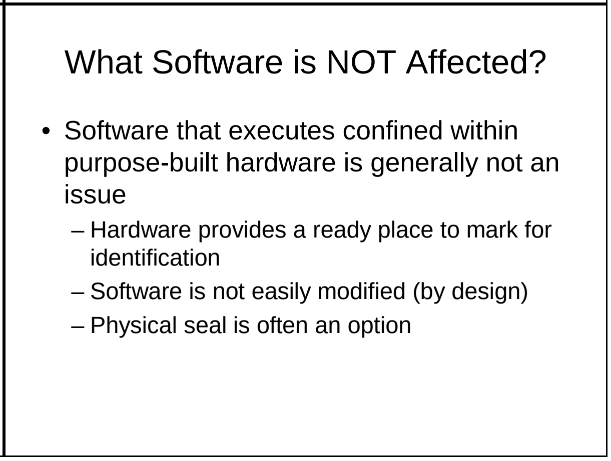
PC-based Software Examples

- Point of Sale Cash Register
- Gas Station Pump Control
- Vehicle Scale In-Out



Point of Conflict?

- Current NTEP policy states that software shall not be separately evaluated and given a CoC
- It could be interpreted that Type Evaluation of the example systems is in conflict with the above rule.
  - No hardware was evaluated in these



What Software is NOT Affected?

- Software that executes confined within purpose-built hardware is generally not an issue
  - Hardware provides a ready place to mark for identification
  - Software is not easily modified (by design)
  - Physical seal is often an option

National Type Evaluation Technical Committee
Software Sector Annual Meeting
March 2-3, 2010   Sacramento, CA

Meeting Summary

Carry-over Items

New Items

Carry-over Items

1. NCWM/NTEP Policies – Issuing CCs for Software

Source: NCWM Reports

Background: For additional background on this item, see the 2009 Software Sector Meeting summary.

Recommendation: 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.

The NTEP committee included this item in their agenda (NTEP Committee 2009 Interim Agenda Item 8); there was no discussion during the open hearing, and this became a Voting item for the 2009 Annual Meeting.
At the 2009 NCWM Annual Meeting, this proposal was passed unanimously by the Conference.

Discussion*:* The NTEP Administrator was asked if there is to be any actual change in any document or is this strictly a procedural change? How do the labs know they can/should handle software items differently now?
The answers to these questions were: there haven't been any changes to Pub 14 this year. The CC's can now say "software." The labs know this; NTEP policy is communicated to the labs. It was suggested that software could be a secondary classification on the certificates.

Conclusions: Our work is complete on this item; it will be removed from the agenda.

2. Definitions for Software Based Devices

Source: 2009 Carryover Item 310-2. This item originated from the NTETC Software Sector and first appeared on the Committee's 2007 agenda as Developing Item Part 1, Item 2.

From NCWM Publication 15, 2010:

310-2 Appendix D – Definition of Electronic Devices, Software-Based and Built-For-Purpose Device

Item Under Consideration:

Delete the current definition of built-for-purpose device as follows:

> ~~*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)*~~

and, add a new definition and a cross-reference to Appendix D in HB 44 for "Electronic devices, software-based" as follows to replace the current definition of "built-for-purpose device":

> ***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*:** For additional background information on this item, please reference the 2009 Software Sector Meeting summary and the 2010 NCWM Interim Meeting Agenda (Pub 15)

At its 2009 Interim Meeting, the CWMA received comments that the proposal is sufficiently developed and recommends moving this item forward as a Voting item on the Committee's agenda. At its 2009 Annual Technical Conference, the WWMA received comments from Mr. Straub, speaking on behalf of SMA, indicating the SMA continues to oppose this item, noting that requirements should apply equally to the two different device types described. The WWMA received no other input on this item and recommends this item should remain Informational until the Software Sector has had an opportunity to review comments from the 2009 NCWM Annual meeting and any comments made at subsequent regional weights and measures association meetings. At its 2009 Annual Meeting, the SWMA recommended keeping the status of this proposal to delete the current definition of built-for-purpose device and add a new definition and a cross-reference to Appendix D in HB 44 for "Electronic devices, software-based" to replace the current definition of "built-for-purpose device" as an Informational item. The SWMA agreed that the Software Sector should continue to work on the proposal until it arrives at some final language. During its 2009 Interim Meeting, NEWMA stated that it supports the Committee's decision to keep this item Informational to allow updated comments from the regional weights and measures associations and other interested parties based on information in the summary of the March 2009 meeting of the Software Sector. Item remains as an informational item on 2010 Annual Meeting

Agenda; the S&T Committee indicated that they look forward to additional work being done on this item by the Sector.

Discussion: Initially it was decided to table discussion on this item; as we worked on items further down the list we would see if it was really necessary to include the 'Type P' and 'Type U' differentiation at this time; if so we would come back and work on the definitions. In particular, Agenda Item 3 (which contained references to the proposed definitions) would be examined in more detail to see if we couldn't satisfy the concerns of the S.M.A. by avoiding differentiation of device types for identification purposes.

Conclusion: When all other agenda items had been discussed it was determined that there was no real need to introduce this differentiation in device types at the current time. It was decided that we would recommend to S&T that ***this item be withdrawn*** for now (with the realization that work on future items may require we reintroduce the concept). The previously proposed language is recorded herein if future requirements would revive the need for the definitions to differentiate between device types.

3. G-S.1. Identification (Software)

Source: NTETC Software Sector

Background: 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. There was much additional comment and various proposed versions of the table from NIST WMD, et al. (The complete background on this item can be seen in the '10 Interim Meeting Agenda NCWM Pub 15, 2010.)

The Sector noted that though currently it is allowable to display the CC number via a menu, there has been some challenges locating this information in the field due to the vagueness of the term "easily recognized." Hence, since it is left to the interpretation of the NTEP laboratory to ascertain whether a device's method for displaying the CC number meets the requirements, this vagueness has not been addressed in this new recommendation.

At the 2009 Software Sector Meeting, it was agreed that the proposed table had not accomplished the intended purpose of clarifying the requirements, indeed it seemed to have generated more confusion. Hence, this item was revisited from the beginning, and it was suggested that a simpler approach be taken, namely to modify the text of G-S.1 to match our intent. The proposal from our Sector was as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect **and manufactured** ~~prior to~~**after January 1, 201X**, shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.

[Nonretroactive as of January 1, 2003]

(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts and ~~***not-built-for-purpose software-based***~~ ***software that is not part of a Type P (built-for-purpose) device.****;*

[Nonretroactive as of January 1, 1968]

(Amended 2003 **and 201X**)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.

[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).

[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier for ~~***not-built-for-purpose***~~ ***software-based electronic*** *devices;*

[Nonretroactive as of January 1, 2004]

(Added 2003) **(Amended 201X)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.

[Nonretroactive as of January 1, 2007]

(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).

[Nonretroactive as of January 1, 2007]

(Added 2006)

(e) an NTEP Certificate of Conformance (CC) number or a corresponding CC Addendum Number for devices that have a CC. The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)

[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003~~, **and**~~**,** 2006 **and 201X**)

G-S.1.1. ~~*Location*~~ *Method of Marking Information for* ~~*Not-Built-For-Purpose*~~ *all Software-Based Devices. –* ***For*** ~~*not-built-for-purpose, software-based*~~ ***devices*** *manufactured* ~~*prior to*~~***after*** *January 1, 201X,* ***either****:*

- *(a) The required information in G-S.1. Identification.* ~~***(a), (b), (d), and (e)***~~ *shall be permanently marked or continuously displayed on the device; or*
- *(b) The Certificate of Conformance (CC) Number shall be:*
  - *(1) permanently marked on the device;*
  - *(2) continuously displayed; or*
  - *(3) accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."*

Note: *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

[Nonretroactive as of January 1, 2004]

(Added 2003) (Amended 2006 **and 201X**)

Discussion: As was noted in the review of what transpired at the Interim Meeting, there appears to be continued resistance, especially from the Scale Manufacturers Association, to differentiating between Type P and Type U software types. From their perspective it is 'all software' and they are concerned that marking requirements will be more complex if we delineate between two different types of software-based devices. Also, the inspectors want to standardize the method of locating the marking information when it is being displayed via menu, and insist that it should be very simple for field personnel to locate. Some additional work by the group resulted in this modified proposal that does not include the new definitions and does not specifically delineate any device types (in fact it removes the existing mention of 'built-for purpose'):

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect ~~**and manufactured after January 1, 201X**~~, shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts ~~*and*~~ ~~*not-built-for-purpose software-based*~~ ~~***software that is not part of a Type P (built-for-purpose) device.***~~*;*
[Nonretroactive as of January 1, 1968]

(Amended 2003 **and 201X**)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier for ~~***not-built-for-purpose***~~ ***software-based electronic*** *devices;*
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 201X)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC. The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.
(Amended 1985, 1991, 1999, 2000, 2001, 2003**, and, 2006** **and 201X**)

Comments: The thinking was that standalone software has no moving or electronic component parts and hence is not required to have a serial number. This was considered acceptable by the Sector; the Sector sees no value in requiring vendors submittals for NTEP approval that are software-only to print serial numbers on their distribution media (CD,DVD, etc). It was observed by CA that if we continue with the concept of only examining 'devices' that typically off-the-shelf PC's have their own serial number, generated by the manufacturer. This can and has been used by the inspectors as a means to meet G-S.1(c) though the prefix/abbreviation is sometimes an issue since the PC manufacturer knows nothing about G-S.1.

It was also suggested that G-S.1.1.b.3 be modified to omit the term "easily recognized"; instead, a limited list of options would be available. A first pass at reworking G-S.1.1(b)(3) resulted in:

G-S.1.1. Location ~~***Method***~~ ***of Marking Information for*** ~~***Not-Built-For-Purpose all***~~ ***Software-Based Electronic Devices. –*** *For* ~~***not-built-for-purpose,***~~ ***software-based*** *devices* ~~***manufactured after January 1, 201X,***~~ *either:*

(a) The required information in G-S.1. Identification. ***~~(a), (b), (d), and (e)~~*** *shall be permanently marked or continuously displayed on the device; or*

(b) The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through ***one or, at most, two levels of access.*** ***~~an easily recognized menu and, if necessary, a submenu.~~*** ***~~Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."~~***

(a) For menu-based systems, "Metrology", "System Identification", or "Help".

(b) For systems using icons, a metrology symbol ("M"), "SI", or a help symbol ("?", "i", or an "i" within a magnifying glass).

Note: *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), (c), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

[Nonretroactive as of January 1, 2004]
(Added 2003) (Amended 2006 **and 201X**)

This new language for G-S.1.1(3)(b) is in the early stages, and the Software Sector would like feedback regarding G-S.1.1(b)(3), particularly suggestions for specific allowed menu items/icons that should be included on the list.

Conclusion**:** The revised G-S.1 (and G-S.1.1) above will be sent to NCWM S&T Committee as our updated recommendation.

[Note: It was observed by WMD (after our meeting adjourned) that there have been several revisions, and revisions to revisions, to our G-S.1 proposals. The proofing (font, bold/italic, etc.) may no longer reflect the correct form with which changes are to be submitted, and they may not actually reflect the changes from what is currently in the 2010 Handbook. This needs to be addressed prior to submission to the S&T Committee; the Chair will compare the proposed language to the current HB44 language and make sure the desired changes are marked properly in the forwarded proposal.]

4. Identification of Certified Software

Source: NTETC Software Sector

Background/Discussion**:** This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?" In previous meetings it was shown that the international community has addressed this issue (both WELMEC and OIML). From WELMEC 7.2:

> **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**.**

From OIML D-31:

> The executable file “**tt100_12.exe**” is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the W&M inspector information that something has changed? (Yes, the Category III audit trail or other means of sealing). How can the W&M inspector identify an NTEP Certified version? (They can’t, without adding additional requirements like what is described here, in conjunction with including the identifier on the CoC).

The Sector believes that we should work towards language that would include a requirement similar to the OIML requirement in HB44. It is also the opinion of the Sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

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

> 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)

Initial draft proposed language: (G-S.1.1?)

Handbook 44 (This has been written into G-S.1.d.3):
Identification of Certified Software:

Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified by the version or revision number~~.~~ ~~The identification~~, and this identification ~~of the software~~ shall be ~~inextricably~~ directly and inseparably linked to the software itself. The version or revision number may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Pub. 14:

Identification of Certified Software:

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.~~

The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

From OIML D-31:

> Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.
>
> The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.

The first sentence of the first paragraph above is already addressed in Handbook 44's marking requirements.

Recommendation: Recommend the following change to Handbook 44, General Code: G-S.1(d) to add a new subsection (3):

(d) the current software version or revision identifier for ***~~not built-for-purpose~~*** ***software-based electronic*** *devices;*
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 201X)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

[Nonretroactive as of January 1, 201X]
(Added 201X)

Also the Sector recommends the following information be added to Pub. 14 as explanation/examples:

- *Unique identifier must be displayable/printable on command or during operation, etc.*
- *At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc (crc32, for example)*

Conclusions: The item needs additional discussion and development by the sector. Outstanding questions: If we allow hard-marking of the software identifier (the Sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to 'inseparably link' the identifier to the software? Do we still have to be able to display/print the identifier if it is hard-marked?

5. Software Protection / Security

Source: NTETC Software Sector

***Background*:** The sector agreed that Handbook 44 already has audit trail and physical seal, but the question on the table is does the Handbook need to be enhanced to sufficiently discourage the facilitation of fraud, intentional or accidental, where software is concerned?

WELMEC and OIML again have addressed this issue specifically when dealing with software. From WELMEC:

> **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 of 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 derived a suitable checklist for Pub 14 from the OIML checklist, and asked the current NTEP labs to begin using this checklist on a trial basis for new type approval applications.

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 all ~~the metrologically significant~~ functions, **designating those that are considered metrologically significant** ~~*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 contro~~l 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** □

The Sector hopes to obtain feedback at this meeting from the NTEP labs regarding this checklist.

Discussion: The labs again indicated they had not had a chance to utilize the checklist. The list was reviewed and some minor modifications to the checklist text were incorporated as shown in this excerpt:

	The software documentation contains:		
		description of all ~~the metrologically significant~~ functions**, designating those that are considered metrologically significant** *~~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** □

Conclusion: Work is ongoing on this item with the intent that it eventually be incorporated as a checklist in Pub 14; again the labs are requested to try utilizing this checklist for any evaluations on software-based electronic devices.

6. 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 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 US W&M requirements.

The Sector **agreed** that the two definitions below for Verified update and Traced update were acceptable.

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.

The Sector also worked towards language proposed for defining the requirements for a Traced Update (currently considered as relevant for Pub 14):

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.

Use of a Category 3 audit trail is acceptable required for the software update logger Traced Update. In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement. If software update is the only loggable event, then the Category 3 audit trail can be limited to only 10 entries. A software update log entry representing a software update shall include the following: the software identification of the newly installed version.

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

A Category III device may include the software update events in the Category III audit log in lieu of a separate software update log; the existing requirement for 1,000 entries supersedes the requirement for 10 entries.

The traceability means and records are part of the metrologically significant software and should be protected as such. If software separation is employed, 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. Manufacturers did indicate that there are methods available to encrypt the audit trail information; however, it cannot be protected from being deleted.)* (include flowchart from OIML D-31)

The Sector discussed how to best move this item forward, and there was also some discussion as to whether new language for the General Code was required. The following new text was proposed:

G-S.9. Metrologically Significant Software Updates

The updating of metrologically significant software shall be considered a sealable event. Metrologically significant software that does not conform to the approved type is not allowed for use.

The NTEP Administrator indicated that the current requirements in G-S.8 already make the statement that any changes that affect metrological function are sealable, hence software updates may be covered and the proposed G-S.9 unnecessary. Todd Lucas suggested we go ahead and submit the proposed G-S.9 to the Committee and request a clarification/interpretation of G-S.8

At the 2009 meeting, the Sector opined that the explicit language proposed for G-S.9 is clearer than any implied requirement in G-S.8. The Sector would like a clarification/interpretation of G-S.8 as it relates to software updates from the S&T Committee (with their response preferably to be included in Pub 16). The Sector will also continue to develop the proposed text (and flow chart) targeted for inclusion in Pub 14.

Discussion: The Sector reviewed the proposal and reconsidered allowing a separate ‘update log’. It was decided that this would probably generate confusion and is not likely to be adopted by manufacturers anyway. Hence, the previously proposed text was modified to require a category III audit trail for ‘traced updates’:

~~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.~~
Use of a Category 3 audit trail is ~~acceptable~~ required for the ~~software update logger~~ Traced Update. ~~In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement.~~ If software update is the only loggable event, then the Category 3 audit trail can be limited to only 10 entries. A ~~software update~~ log entry representing a software update shall include the ~~following:~~ the software identification of the newly installed version.

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

~~A Category III device may include the software update events in the Category III audit log in lieu of a separate software update log; the existing requirement for 1,000 entries supersedes the requirement for 10 entries.~~

Conclusions: The general consensus of the group after considering feedback from external interested parties is that a new G-S.9 with explicit requirements is not necessary (nor likely to be adopted by the Conference) and that this requirement belongs in the Pub. 14 lists of sealable parameters rather than in Handbook 44; i.e.

The updating of metrologically significant software shall be considered a sealable event.

Additional work is to be done to further develop the proposed text toward inclusion in Pub 14.

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

Source: NTETC Software Sector

Background 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)

Some discussion about system information requirements for the inspector took place… does the inspector really need to have access to OS, RAM information, etc? (General opinion seems to be if there is a dependency, then the NTEP lab would specifically include that requirement in the CoC.)

Audit trail info – the question was asked, does there need to be a specific requirement for providing access to this information?

Regarding the concept of First Final – There was some concern expressed as to how the inspectors are able to discern where the indication of first final be found for the SYSTEM (as opposed to the DEVICES in the system). What devices in the system are of concern to the inspector? The NTEP Administrator indicated that field inspectors need to follow the system all the way to receipt/bill generation.

Data transmission is an issue when considering systems as opposed to devices… how far does the inspector's jurisdiction extend? (Should we model future requirements on the WELMEC section concerning DTD/DSD?)
Decision: data transmission/storage is not currently being addressed by the Sector at this time.
Since part of the Sector's mission is education, do we want to assist in developing training aids for labs/inspectors related to evaluating/inspecting software-based devices? This will be a topic to be added to the Sector's agenda for the next meeting.

At the 2009 meeting, the Sector decided to continue to develop this item, and initiate a new agenda item specific to inspector training in relation to evaluating/validating software-based devices.

Discussion: A question from the floor requested opinion as to whether this agenda item continued to serve a purpose. During discussion, it was stated that the goals of this item have all been addressed as part of all the other agenda items save one (training), and inspector training will now be covered in a new item (Training of Field Inspectors), leaving this item without merit.

Conclusion: No argument was made for retaining this item as a separate item on the agenda. This item will be removed from future agendas.

8. NTEP Application for software ~~requiring a separate Certificate of Conformance –~~based electronic devices

Source: NTETC Software Sector

Background/Discussion: The purpose of initiating this item was to identify issues, requirements and processes for type approving type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. Question: what gets submitted? What requirements and mechanisms for submission should be available?

Validation in the lab **-** all required subsystems shall be included to be able to simulate the system as installed.

It was noted this agenda item is irrelevant if the NTEP Committee does not approve the pending item up for vote.

John Roach (CA NTEP Lab) stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale labs and scale manufacturers indicated that this is not usually done for scale evaluations.

Conclusion of 2009 Sector Meeting: The Sector will continue to develop this item, contingent on the status of the related NTEP Committee agenda item after the 2009 Annual meeting.

Discussion: Since the NTEP committee passed the related item at the Annual we will continue to work on this.

The NTEP director indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

Conclusion: The item will be revisited at the 2011 Meeting and it will be decided whether to begin further development of this item at this time.

New Items

9. Training of Field Inspectors

Source: NTETC Software Sector

Background: During discussions at the 2009 meeting, the Sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

Discussion: CA has an EPO (Examination Procedure Outline) that begins to address this. Use Handbook 112 as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in HB 44.
- Reference materials / information sources
- Safety

System Verification Tests

NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. **G-S.1 (1.10)**

1.1. Manufacturer.

1.2. Model designation.

2. Provisions for sealing. **G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]**

2.1. Verify sealing category of device (refer to Certificate of Approval for that system).

2.2. Verify compliance with certificate.

3. Units of measure.

3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. **G-S.5.2.2(a); G-S.5.1 [1.10]**

3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.

4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).

4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.

5. Indications and displays.

5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.

6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load.

S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4

6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications.

S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4

7. Behind zero indication.

7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the

scale. **S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2**
Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.
7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.
8. Over capacity.
8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. **S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]**
8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.
Measuring Devices
10. Motion detection.
10.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. **S.2.4.1. (3.30)**
11. Over capacity.
11.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value**.**
NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

Conclusion: This item is in the early stages; work will continue on the item working toward materials to aid in the training of field inspectors. It was indicated that working in conjunction with the Professional Development Committee to develop training materials, etc. would be a logical path of progress once we have developed the information content to include.

10. Next meeting

Background: The Sector is on a yearly schedule for Sector meetings. The NTEP Administrator determines when the next meeting is possible.

Discussion: The NTEP Administrator indicated that the NTETC meetings are to be scheduled where the conference gets the most 'bang for the buck', so that implies (considering our Spring schedule) one of the states with an NTEP lab. Hence we've been rotating among Annapolis, Columbus, and Sacramento. It was also mentioned by the Technical Advisor that this rotating of the location has been quite beneficial to the group, considering the variety of input from individuals not typically able to make the trip to attend distant meetings.

Conclusion: Given the above, it was suggested that it would be Maryland's turn in 2011. In keeping with the March timeframe and trying to avoid the last blast of winter, the group decided to return to Annapolis, preferably March 15-16th, 2011. Second choice would be the following week (March 22nd - 23rd). The Maryland lab personnel will assist the NCWM staff in suggesting one or more suitable host facilities for the meeting.

Appendix A: Report on 2009 Interim Meeting

There were two items on the NCWM Specifications and Tolerances committee agenda related to our mission – 310-2 (definitions of software based devices) and 310-3 (marking requirements). The consensus was that they still need work, and they remain “informational.”

It seemed from the comments made during the open hearings that the membership didn’t see a clear benefit to the field inspectors, and the scale manufacturers were also resistant to the change, fearing distinction between different types of devices would complicate marking, and additionally the SMA didn’t see a difference between built-for-purpose and non-built-for-purpose.

In general, the feedback at the Interim gave the impression to Sector members that attended that we need to back up a little.

Appendix B: Report on International W&M Activity

There's a new project regarding field verification, but there likely won't be activity this year.

There weren't too many changes to WELMEC 7.2. They are mainly clarifications. The current methodologies are now considered a bit too restrictive, so they're being reconsidered.

There has been an update to one of our referenced WELMEC documents since our last Software Sector meeting:

> *Software Guide (Measurement Instruments Directive 2004/22/EC)* is now at Issue 4.

You can download an updated copy of this document at http://www.welmec.org/publications/7-2.asp

The changes are minor, including:

- Removal of the requirement that the NB maintain a file of the documentation and (if necessary) the software supplied for Type P & Type U submissions.
- Software Download extension has two additions, listed below in blue below:

9 Extension D: Download of Legally Relevant Software
This extension shall be used for the download of legally relevant software as long as the metrological characteristics remain unchanged and the declaration of conformity is still valid, e.g. bug-fixes. These requirements are to be considered in addition to the basic requirements for Types P and Type-U described in Chapters 4 and 5 in the guide.

D2: Authentication of downloaded software
Means shall be employed to guarantee that the downloaded software is authentic, and to indicate that the downloaded software has been approved by an NB.
Specifying Notes:
1. Before the downloaded software is used for the first time, the measuring instrument shall automatically check that:
a. The software is authentic (not a fraudulent simulation).
b. The software is approved for that type of measuring instrument.
2. The means by which the software identifies its NB approval status shall be made secure to prevent counterfeiting of the NB status.
3. If downloaded software fails any of the above tests, see D1.
4. If a manufacturer intends to change or update the legally relevant software he shall announce the intended changes to the responsible notified body. The notified body decides whether an addition to the existing TEC is necessary or not. For software download it is indispensable that there is a software identification which is unambiguously assigned to the approved software version.

Appendix C: Final Attendee List

William Arce
USDA, GIPSA, PSP
210 Walnut Street, Room 317
Des Moines, IA 50309
P. (515) 323-2510 **F.** (515) 323-2590
E. william.arce-arana@usda.gov

Dennis Beattie
Measurement Canada
400 St. Mary Ave
Winnipeg, Manitoba R3C 4K5
Canada
P. (204) 983-8910 **F.** (204) 983-5511
E. dennis.beattie@ic.gc.ca

Doug Bliss
Mettler-Toledo, Inc.
1150 Dearborn Drive
Worthington, OH 43085
P. (614) 438-4307 **F.** (614) 438-4355
E. doug.bliss@mt.com

Dick Dirksen
Vande Berg Scales
770 7th St. NW
Sioux Center, IA 51250
P. (712) 722-1181

Michael Frailer
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21224
P. (410) 841-5790 **F.** (410) 841-2765
E. fraileml@mda.state.md.us

Andrew Gell
FOSS North America, Inc.
8091 Wallace Road
Eden Prairie, MN 55344
P. (952) 974-9892
E. agell@fossnorthamerica.com

Teri Gulke
Liquid Controls
105 Albrecht Drive
Lake Bluff, IL 60044-2242
P. (847) 283-8346 **F.** (847) 295-1170
E. tgulke@idexcorp.com

Norman Ingram
CDFA / DMS
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828-1812
P. (916) 229-3016 **F.** (916) 229-3015
E. ningram@cdfa.ca.gov

Michael Kelley
Ohio Department of Agriculture
Division of Weights and Measures
8995 East Main Street
Reynoldsburg, OH 43068
P. (614) 728-6290 **F.** (614) 728-6424
E. mkelley@agri.ohio.gov

Dan Parks
California Division of Measurements Standards
6790 Florin Perkins Rd., Suite 100
Sacramento, CA 95828
P. (916) 229-3000 **F.** (916) 229-3015
E. dparks@cdfa.ca.gov

Paul A. Lewis, Sr.
Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868-2404
P. (715) 434-5322 **F.** (715) 234-6967
E. plewis@ricelake.com

James Pettinato
FMC Technologies Measurement Solutions, Inc.
1602 Wagner Avenue
Erie, PA 16510
P. (814) 898-5250 **F.** (814) 899-3414
E. jim.pettinato@fmcti.com

Joe Raspino
California Division of Measurement Standards
6790 Florin Perkins Rd., Suite 100
Sacramento, CA 95828
P. (916) 229-3070 **F.** (916) 229-3015
E. jraspino@cdfa.ca.gov

Dan Reiswig
California Division of Measurement Standards
6790 Florin Perkins Rd., Suite 100
Sacramento, CA 95828
P. (916) 229-3023 **F.** (916) 229-3015
E. dreiswig@cdfa.ca.gov

John Roach
California Division of Measurement Standards
Type Evaluation Program
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916) 229-3014 **F.** (916) 229-3015
E. jroach@cdfa.ca.gov

Brett Saum
San Luis Obispo County Weights and Measures
2156 Sierra Way, Suite A
San Luis Obispo, CA 93401
P. (805) 781-5922 **F.** (805) 781-1035
E. bsaum@co.slo.ca.us

Matthew Stevens
California Division of Measurement Standards
Type Evaluation Program
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916) 229-3018 **F.** (916) 229-3015
E. mstevens@cdfa.ca.gov

Ambler Thompson
NIST, Weights & Measures Division
100 Bureau Drive, MS 2600
Gaithersburg, MD 21701
P. (301) 975-2333 **F.** (301) 975-8091
E. ambler@nist.gov

Van Thompson
California Division of Measurement Standards
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916) 229-3025 **F.** (916) 229-3016
E. vthompson@cdfa.ca.gov

James Truex
National Conference on Weights and Measures
88 Carryback Drive
Pataskala, OH 43062
P. (740) 919-4350 **F.** (740) 919-4348
E. jim.truex@ncwm.net

Appendix D

National Type Evaluation Technical Committee (NTETC) Software Sector Meeting Summary

March 15-16, 2011 / Annapolis, Maryland

INTRODUCTION

The charge of the National Type Evaluation Technical Committee (NTETC) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the National Type Evaluation Program (NTEP) Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be nonretroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content **NTEP - D Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OIML	International Organization of Legal Metrology
CC	Certificate of Conformance	PDC	Professional Development Committee
GMMs	Grain Moisture Meters	S&T	Specifications and Tolerances Committee
NCWM	National Conference on Weights and Measures	SMA	Scale Manufactures Assocation
NTEP	National Type Evaluation Program	WELMEC	European Cooperation in Legal Metrology
NTETC	National Type Evaluation Technical Committee		

Details of All Items
(In order by Reference Key)

CARRY-OVER ITEMS

1. Software Identification / Markings

Source:
NTETC Software Sector

Background / Discussion:
Since its inception the sector has wrestled with the issue of software identification and marking requirements. For more background information on this item, see the 2010 NTETC Software Sector Meeting Summary and the 2011 NCWM Interim Meeting S&T Committee Agenda Item 310-2.

On the first day of discussion, the sector agreed that the revisions to G-S.1 and G-S.1.1 as they were presented in the 2010 *NCWM Publication 15* and as Informational Items in *NCWM Publication 16* still required some clarification in certain areas. There seems to be confusion regarding requirements for purely mechanical devices now, and there is no indication of the preference for hard-marking when the option to mark or display is allowed.

Feedback received from NCWM membership seems to indicate a preference to not delineate between device types (at least where marking requirements are concerned). This was taken into account as the sector reviewed the current and previously proposed language.

In general, the sector agreed that for the purposes of marking there was no reason to distinguish between different types of software (i.e. a software on a CD that is to be installed on a computer, or software embedded in a chip within a built-for-purpose device).

The following draft revision of the language in G-S.1 and G-S.1.1 was crafted to try to address some of these concerns, and as a basis for further discussion:

NIST Handbook 44

G-S.1. Identification. – All equipment, except weights~~, and~~ separate parts necessary to the measurement process but not having any metrological effect~~*, and software-based devices covered in G-S.1.1. Location of Marking Information*,*~~ shall be clearly and permanently marked **as per G-S.1.1.** for the purposes of identification with the following information:
*[*Nonretroactive as of January 1, 20XX]*

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts ~~**and not-built-for-purpose software-based software devices;**~~
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) when metrologically significant software is employed, the current software version or revision identifier, which shall be directly and inseparably linked to the software itself ~~**for not-built-for-purpose software-based electronic devices;**~~

[Nonretroactive as of January 1, 2004]

(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC.

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003, and, 2006 and 201X)

**added by S&T Committee based on SMA comments, and not in original NTETC Software Sector submission*

The clause regarding making the software version / revision inseparably linked to the software may or may not be included in current recommendations. Feedback will be obtained from the Scale Manufactures Association (SMA) in April.

G-S.1.1. Location of Marking Information ~~for Not-Built-For-Purpose all Software-Based Devices. – For not-built-for-purpose, software-based devices, either:~~

(a) The required information in G-S.1. Identification~~. (a), (b), (d), and (e)~~ shall be permanently marked or continuously displayed on the device; or

(b) The required information in G-S.1. Identification shall be available via the user interface. The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through ***one or, at most, two levels of access.*** ~~***an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."***~~

(i) For menu based systems, "Metrology," "System Identification," or "Help."

(ii) For systems using icons, a metrology symbol "(M)", "(SI)," or a help symbol ("?," "i," or an "i" within a magnifying glass).

Note: *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*
[Nonretroactive as of January 1, 2004]
(Added 2003) (Amended 2006 **and 20XX**)

For several years now, the sector has been recommending updates to G-S.1. to eliminate the use of the undefined term "not built for purpose", and to add a requirement for marking software version/revision for ALL devices using metrologically significant software (currently built-for-purpose devices are excluded from this requirement). However, the sector tried to fit it in by generalizing other areas of the Code (meaning G-S.1.1) to apply to all software-based devices.

At the 2010 NCWM Interim Meeting, Mr. Truex, NTEP Administrator, provided a history of how this issue evolved. He noted that there were multiple attempts to address software in not-built-for purpose devices. The

NTETC Software Sector has attempted to further simplify the identification requirements that apply to software-based systems and has made multiple suggestions that were not accepted. The sector has taken a step back and is trying to get the point across that the marking requirements are not for the manufacturer, but to assist the inspector in the inspection process and in assessing whether or not a specific device, including software, is covered under an NTEP Certificate of Conformance (CC). The sector realizes that ideally this information is not going to be physically marked on the device and is looking for alternatives in which this information can be provided electronically to inspectors in an easily accessible manner. It will likely be provided on the device's display screen and there is limited space for this information to be displayed. The sector is looking for input on the general direction it should take in developing/updating *NIST Handbook 44* requirements. If the direction seems reasonable, the sector will further develop the idea; if not, the sector will consider an alternative direction.

Comments in response to that question posed to the Conference indicated that the sector was on the right track; but the language needed additional work. Limiting the options for locating required marking information seemed to be a well-received idea.

Further discussion regarding "easily recognizable" was addressed previously with an initial list of menu options/icons that would act as the "defining" set of acceptable menu selections/icons for finding the CC number of the device. The idea was to limit the options to a finite set, thus assuring evaluators and field inspectors had at least a reasonable limit to the possible ways to obtain this information. There was good feedback and discussion from other groups and was considered during the 2011 NTETC Software Sector Meeting, and the sector did modify the document to eliminate some options that were deemed problematic, etc. Originally the plan was to put this into *NIST Handbook 44* but further discussion at the 2011 NTETC Software Sector Meeting led to the consensus that the existing language is sufficient, and using such a list as guidance for the evaluating laboratories (e.g. *NCWM Publication 14*) would be the proper approach. Hence, the list of menu text/icons as updated to reflect the comments received can be re-circulated, but is now the opinion of the sector that this list is best targeted at *NCWM Publication 14*.

Table 1 – NTETC Software Sector Proposed Menu Text / Icons

Permitted Menu Text Examples	Permitted Icon Shape Examples	Essential Characteristics
Information Info	i i i	Top level menu text or icon • Icon text is a lower case "i" with block serifs • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Help ?	? ?	Top level menu text or icon • Icon text is a question mark • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Metrology Metrological Information	M M	Top or second level menu text or icon • Icon text is an upper case "M" • Text color may be light or dark but must contrast with the background color • Icon may have a rectangle or rounded rectangle border. Note (2011 mtg): using a rectangle is problematic because it matches a symbol used in Europe. A circle would be preferred. Green M may also be an issue due to it being used as a metrology mark in EU. • If present, the activation of this menu text/icon must recall at a minimum the NTEP CC number.
SI S.I. System Information System Info	SI SI	Top or second level menu text or icon • Icon text is upper case "SI" • Text color may be light or dark but must contrast with the background color • Icon may have a rectangle or rounded rectangle border • If present, the activation of this menu item/icon must recall at a minimum the NTEP CC number. • The SI is problematic since it is also used to identify the International System of Units.
NTEP Data N.T.E.P. Certificate	NTEP	This one is debatable – what if the certificate is revoked? Does NTEP grant holders of CCs the right to display the logo on the device, or just in documentation?

Acceptable examples of where the text or icon may be displayed:

1. The "M" icon is available on the home screen. Activation of the icon displays a new screen containing the CC number and some additional metrology information including the software version/revision number(s).
2. The "SI" icon is available on the home screen. Touch screen activation of the icon displays a pop-up containing the CC number. Releasing the icon erases the pop-up.
3. The main screen contains the "i" icon (information). Activating this icon displays a screen of other icons including the "M" icon. Activating the "M" icon displays the NTEP CC.
4. The main menu includes a "Help" selection which in turn contains a "Metrology" selection. Activation of the Metrology selection displays a pop-up screen containing all global metrological approvals, including the NTEP CC number. The user manually dismisses the pop-up screen by pressing the [X] button.

The main menu includes an “Info” selection which in turn contains a “SI” selection. Activation of the SI selection displays a pop-up screen containing all global metrological approvals, including the NTEP CC number. The user manually dismisses the pop-up screen by pressing the [OK] button.

Comments from NTETC Weighing Sector Comments:
The NTETC Weighing Sector reviewed the initial list of menu text and icons and provided the following comments:

- Mr. Flocken, Mettler-Toledo, Inc., indicated that the green M is an EU metrology mark and for that reason should not be considered an acceptable icon.
- There was general consensus amongst NTETC Weighing Sector members that the SI should not be considered acceptable since it is also used to identify the International System of Units.

Comments from NTETC Measuring Sector Comments:
The NTETC Measuring Sector had no additional technical guidance to offer to the S&T Committee on this issue. However, based on comments from sector members present, the NTETC Measuring Sector expressed general support for trying to refine the marking requirements and limit the number of options for marking keys that enable the inspector to view the required marking information.

Potential additions to the list of acceptable options would be an icon or menu option showing “W/M”, “W&M”, or “Menu” for a top level menu text option.

NTETC Grain Analyzer Sector Comments:
The NTETC Grain Analyzer Sector found the wording of G-S.1.1. confusing. It seemed to say that the markings spelled out in G-S.1. were to be EITHER permanently marked or continuously displayed on the device OR the Certificate of Conformance (CC) Number shall be either: permanently marked or continuously displayed, or accessible through menu or icon. To some, this implied that the software version identifier did NOT have to be displayed. Others believed that the “OR” phrase meant that only the CC had three options for marking (permanent, continuously displayed, or accessible via menu or icon), and that the software/firmware version/revision number must be either permanently marked or continuously displayed.

Regardless of how the wording is interpreted, the NTETC Grain Analyzer Sector agreed that it was not practical to permanently mark or continuously display the software/firmware version/revision identifier for Grain Moisture Meters (GMMs). The sector recommends that G-S.1.1.(b) be amended to include accessing the software version or revision identifier by menu or icon. At present all NTEP GMMs are built-for-purpose. They all have permanently marked CC numbers. Software version/revision identifiers, however, are accessible by menu or icon. GMM displays are of limited size. Some existing devices don’t have room to display the software version/revision identifier on every “screen”. Hard marking of that identifier is not practical, because it precludes updating software without also replacing the hard-marked label.

SMA Comments:
The SMA supports the requirement to access a version number for software based devices. The SMA looks forward to the NTETC Software Sector's definition of the term “software based device”.

SMA opposed the definition we provided previously. From the 2009 NTETC Software Sector Meeting Summary and 2010 *NCWM Publication 15* 310-2:

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.

The sector's previous efforts to incorporate these concepts into the text of G-S.1 seemed to result in confusion and concern over unintended side effects of the changes proposed, and hence met with resistance. This led the sector to consider a new approach. Rather than modify a broadly applicable section of general code language to address software concerns, the idea of inserting specific concerns as new clauses seemed much less likely to cause unintended changes (side effects).

This topic was again discussed on the second day of the meeting, resulting in the following proposed new language for G-S.1 and G-S.1.1 that contains modifications that are less invasive and more specific to the intent of the sector:

G-S.1. Identification. – All equipment, except weights, and separate parts necessary to the measurement process but not having any metrological effect, shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts ~~**and not-built-for-purpose software-based software devices;**~~
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) **when metrologically significant software is employed,** the current software version or revision identifier ~~**for not-built-for-purpose software-based electronic devices;**~~
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 201X)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be accessible via the display in lieu of being permanently marked. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:

(a) The user interface does not have any control capability to activate the indication of the version or revision identifier on the display, or the display does not technically allow the version or revision identifier to be shown (analog indicating device or electromechanical counter) or

(b) the device does not have an interface to communicate the version or revision identifier or

(c) after the production of the device a change of the software is not possible, or only possible if the hardware or a hardware component is changed.

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC.

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.
(Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~and,~~ 2006 **and 201X**)

G-S.1.1. Location of Marking Information for ~~Not-Built-For-Purpose~~ all Software-Based Devices. *– For ~~not-built-for-purpose,~~ software-based devices, either:*

(a) The required information in G-S.1. Identification. ~~(a), (b), (d), and (e)~~ shall be permanently marked or continuously displayed on the device; or

(b) The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through **one or, at most, two levels of access.** ~~**an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."**~~

(i) For menu based systems, "Metrology," "System Identification," or "Help."

(ii) For systems using icons, a metrology symbol "(M)", "(SI)," or a help symbol ("?," "i," or an "i" within a magnifying glass).

> ***Note:*** *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) (Amended 2006 **and 201X**)

Note: the striking of some of the text in G-S.1(c) should NOT in the opinion of the sector result in an interpretation that it is a requirement to mark a serial number on standalone software. Standalone software has no moving or electronic parts and hence is already exempt from the requirement.

The new language in G-S.1.1 reflects that the sector reached consensus on the following positions:

- The software version/revision should (with very few exceptions – see D-31 5.1.1) be accessible via the user interface.
- The means by which the software version is accessed must be described in the CC.

In addition, it was asserted that the previously recommended changes to G-S.1.1 (b)(3) in fact are not really necessary; the current language of *NIST Handbook 44* empowers the labs to enforce "easily recognizable" as they see fit. In fact, the previously generated "list" of icons and menu options could certainly be used by the examining lab as part of the approval process (e.g. in *NCWM Publication 14*). Of course, a manufacturer who is reviewing *NIST Handbook 44*so as to develop an acceptable device may benefit from more explicit guidance. Where does such guidance belong?

Comments related to the circulated list included a comment from the SMA suggesting that a definition is needed for a "software-based devices". SMA opposed the definitions previously put forth by the sector. It was suggested that perhaps the SMA would be more amenable to a definition that doesn't differentiate between software types.

Additional discussion on the topic of G-S.1 was related to the following concept, which may eventually result in additional recommendations to amend G-S.1:

The sector sees merit to requiring some "connection" between the software identifier (i.e., version/revision) and the software itself (as does International Organization of Legal Metrology (OIML), see D-31). The proposal being considered is to add a new subparagraph to G-S.1.(d) to read as follows (with the expectation that examples of acceptable means of implementing such a link would be included in *NCWM Publication 14*).

> "The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software."

Conclusion:
The sector wishes to obtain feedback on the newly recommended language for G-S.1. and G-S.1.1. since it does deviate somewhat from previous submissions. It is hoped that the various interested sectors, regions and associations will give this new proposal careful thought and submit their concerns to the NTETC Software Sector.

The list of suggested icons/menus that should be considered finite options for manufacturers was updated to reflect comments received by the sector. The sector now believes this approach is adequate without a change to *NIST Handbook 44*; the NTEP laboratories would be able to enforce "easily recognizable" against this finite list. Hence, the sector recommends the list be inserted into *NCWM Publication 14.*

As to the requirement to have some "connection" between the software identifier and the software itself, the sector felt that this topic requires more work, so it will be split out into a separate item and put forth as a separate proposal.

Crafting a definition for "software based device" may be included as an item in a future agenda. Note the term "not built for purpose, software based device" is already used in *NIST Handbook 44.*

2. Identification of Certified Software

Source:
NTETC Software Sector

Background / Discussion:
This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?" In previous meetings it was shown that the international community has addressed this issue (both European Cooperation in Legal Metrology (WELMEC) and OIML).

From WELMEC 7.2:

> **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.

From OIML D-31:

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the weights and measures inspector information that something has changed? Yes, the Category III audit trail or other means of sealing.

How can the weights and measures inspector identify an NTEP Certified version? They can't, without adding additional requirements like what is described here, in conjunction with including the identifier on the CC.

The sector believes that we should work towards language that would include a requirement similar to the OIML requirement in *NIST Handbook 44*. It is also the opinion of the sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

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

From OIML:

> 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)

Initial draft proposed language: (G-S.1.1?)

NSIT Handbook 44 (This has been written into G-S.1.d.3): Identification of Certified Software:

> Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified by the version or revision number~~. The identification,~~ and this identification ~~of the software~~ shall be ~~inextricably~~ directly and inseparably linked to the software itself. The version or revision number may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

NCWM Publication 14:

> **<u>Identification of Certified Software:</u>**
>
> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~**The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.**~~
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

From OIML D-31:

> Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.
>
> The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.

The first sentence of the first paragraph above is already addressed in *NSIT Handbook 44*'s marking requirements.

In 2010, the sector crafted a draft recommendation for the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

> *(d) the current software version or revision identifier for* ~~***not built for purpose***~~ ***<u>software-based electronic</u>*** *devices;*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) **<u>(Amended 20XX)</u>**
>
> > *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> > *[Nonretroactive as of January 1, 2007]*
> > *(Added 2006)*

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

There was some additional discussion on this item regarding where this new requirement was best located. It was suggested that the first sentence of G-S.1.d. (3) could be added as a clause to the base paragraph G-S.1. (d) text, e.g. " *the current software version or revision identifier for* ~~**not built-for-purpose**~~ ***software-based*** *devices, which shall be directly and inseparably linked to the software itself*;" .

It also was suggested that the second sentence in G-S.1.d. (3) might be more suitable for *NCWM Publication 14*, as it describes more 'how' than 'what' the requirement entails.

In addition, the sector considered the following information to be added to *NCWM Publication 14*as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions that are still outstanding:

5. If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to "inseparably link" the identifier to the software?
6. If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

Conclusion:
The item needs additional discussion and development by the sector. It is hoped that the sector will obtain some feedback regarding *NCWM Publication 14* recommendations from the SMA in April, and other sectors, regions and interested parties.

3. Software Protection / Security

Source:
NTETC Software Sector

Background / Discussion:
The sector agreed that *NIST 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 of development techniques have been applied.
>
> This requirement includes consideration of:
>
> 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 *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussion at October 2007 NTETC Software Sector Meeting.

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 laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In Feb ruary 2011, the North Carolina laboratory was also given a copy of the check list to try.

1. Devices with Embedded Software TYPE P (aka built-for-purpose)

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **AND** □ Yes □ No □ N/A

1.2. 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.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

2. 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)

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. 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

3. Software with Closed Shell (no access to the operating system and/or programs possible for the user)

3.1. 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

3.2. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User

4.1. 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 Weights and Measures jurisdiction and type-specific parameters). □ Yes □ No □ N/A

4.2. 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

5. Software Interface(s)

5.1. Verify the manufacturer has documented:

5.1.1. The program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.1.2. The protective software interface itself is part of the metrologically significant software.

5.1.3. The functions of the metrologically significant software that can be accessed via the protective software interface. □ Yes □ No □ N/A

5.1.4. The parameters that may be exchanged via the protective software interface are defined. □ Yes □ No □ N/A

5.1.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.1.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

The laboratories were polled to obtain any feedback on the use of the checklist.

The Maryland laboratory attempted to use this checklist a few times. Mr. Payne, Maryland Department of Agriculture, had some difficulty obtaining answers from the manufacturers because the individual(s) interacting with Mr. Payne did not always have the required information on hand. More experience in using the checklist will help determine what needs to be revised.

It was suggested that the checklist could be sent to manufacturers for their feedback as well, with the stipulation that it a completely voluntary exercise and purely informational at this point. The laboratories will coordinate with willing manufacturers to obtain feedback.

Conclusion:
Work is ongoing on this item with the intent that it eventually will be incorporated as a checklist in *NCWM Publication 14*; again the laboratories are requested to try utilizing this checklist for any evaluations on software-based electronic devices.

4. Software Maintenance and Reconfiguration

Source:
NTETC Software Sector

Background / Discussion:
After the software is completed, what do the manufacturers use to secure their software? At the 2010 NTETC Software Sector Meeting, significant discussion on the approach taken by OIML were reviewed by the sector.

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

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 can contain both means of security

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

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 weights and measures requirements.

The sector agreed that the two definitions below for were acceptable.

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.

The sector discussed how to best move this item forward, and there was also some discussion as to whether new language for the General Code was required. The following new text was proposed:

G-S.9. Metrologically Significant Software Updates. - The updating of metrologically significant software shall be considered a sealable event. Metrologically significant software that does not conform to the approved type is not allowed for use.

Mr. Truex, NTEP Administrator, indicated that the current requirements in G-S.8 already make the statement that any changes that affect metrological function are sealable, hence software updates may be covered and the proposed G-S.9 unnecessary. Mr. Lucas, Ohio Department of Agriculture, suggested the sector go ahead and submit the proposed G-S.9 to the committee and request a clarification/interpretation of G-S.8

At the 2009 meeting, the sector opined that the explicit language proposed for G-S.9 is clearer than any implied requirement in G-S.8. The sector would like a clarification/interpretation of G-S.8 as it relates to software updates from the S&T Committee (with their response preferably to be included in *NCWM Publication 16*). The sector will also continue to develop the proposed text (and flow chart) targeted for inclusion in *NCWM Publication 14*.

The sector reviewed the proposal and reconsidered allowing a separate "update log". It was decided that this would probably generate confusion and is not likely to be adopted by manufacturers anyway. Hence, the previously proposed text was modified to require a Category III audit trail for "traced updates":

> ~~**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.**~~
>
> Use of a Category 3 audit trail is ~~**acceptable**~~ required for the ~~**software update logger**~~ **Traced Update**. ~~**In this case the existing requirement of 1,000 entries supersedes the 10 entry requirement. If software update is the only loggable event, then the Category 3 audit trail can be limited to only 10 entries.**~~ A ~~**software update**~~ log entry representing a software update shall include the ~~**following: the**~~ software identification of the newly installed version.
>
> - ~~**An event counter;**~~
> - ~~**the date and time of the change; and**~~
> - ~~**the event type/parameter ID, which indicates a software update event (if not using a dedicated update log);**~~
> - ~~**the new value of the parameter, which is the software identification of the newly installed version.**~~
>
> ~~**A Category III device may include the software update events in the Category III audit log in lieu of a separate software update log; the existing requirement for 1,000 entries supersedes the requirement for 10 entries.**~~

In 2010, the general consensus of the sector after considering feedback from external interested parties is that a new G-S.9. with explicit requirements is not necessary (nor likely to be adopted by the Conference) and that this requirement belongs in *NCWM Publication 14* lists of sealable parameters rather than in *NIST Handbook 44*; i.e.

> **The updating of metrologically significant software shall be considered a sealable event.**

Additional work is to be done to further develop the proposed text toward inclusion in *NCWM Publication 14*.

Since the 2010 NTETC Software Sector Meeting, the NTETEC Grain Analyzer Sector remitted the following: At its August 2009 NTETEC Grain Analyzer Sector Meeting the sector questioned the need for a definition of "Traced Update". The traced update was initially intended to cover cases in Europe where the National Body controls a network of devices and wants to update all the devices simultaneously from a central location. Denmark and France do this with NIR Grain Analyzers. Even though individual states may still require that a device updated via a "Traced Update" must be "returned to service" by a registered serviceperson before it can be used, the sector may want to consider adopting "Traced Update" requirements for all Category 3 Grain Analyzers. The device is still subject to later inspection by state weights and measures personnel. By designing to the requirements for "traced update", states might be encouraged to allow devices updated to those requirements to be returned to service without requiring a visit by a registered serviceperson. No formal comments or recommendations were made by the NTETC Grain Analyzer Sector.

The NTETC Software Sector concurred that these definitions should be included in *NCWM Publication 14* in the section where they are used (since *NCWM Publication 14* does not have a separate section devoted strictly to definitions).

It's possible that the Philosophy of Sealing section of *NCWM Publication 14* may already address the above if the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit:

> **Use of a Category 3 audit trail is required for a Traced Update. A log entry representing a traced software update shall include the software identification of the newly installed version.**

Conclusion:
It seemed sensible to recommend consolidating the definitions with the above statement and placing them into *NCWM Publication 14*. The sector recommends the following:

> **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 Category 3 audit trail. The audit trail entry shall include the software identification of the newly installed version.
>
> **The updating of metrologically significant software shall be considered a sealable event. The software that checks for authenticity and integrity for a Traced Update, as well as the software responsible for generating and viewing the audit trail, is metrologically significant.**

5. NTEP Application for Software and Software-based Devices

Source:
NTETC Software Sector

Background/ Discussion:
The purpose of initiating this item was to identify issues, requirements and processes for type approving type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the lab - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. (Clarification at 2011 NTETC Software Sector Meeting - this has never been required for type approval, except for retail motor fuel distribution systems.) Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting the sector will continue to work on this. Mr. Truex, NTEP Administrator, indicated that the sector can move in this direction, but felt that it was somewhat premature to develop this thoroughly in 2010. At the point where the sector has developed checklist requirements, then it could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. The description of this agenda item was modified as shown in the marked up heading.

At the 2010 NTETC Software Sector Meeting, it was decided that this item would be revisited at the 2011 meeting and it will be decided whether to begin further development of this item at that time.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5. states, "The approval applicant is responsible for the provision of all the required equipment and components." This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4):

- A description of the software functions that are metrologically significant, meaning of the data, etc.
- A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).
- A description of the user interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.

Conclusion:
These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop *NCWM Publication 14* requirements is needed, after more input from the labs is gathered.

6. Training of Field Inspectors

Source:
NTETC Software Sector

Background / Discussion:
During discussions at the 2009 NTETC Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. It was suggested that the sector could use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources
- Safety

System Verification Tests
NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.

2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2
   Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.
   7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.
8. Over capacity.
   8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]
   8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.
   9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)
10. Over capacity.
    10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

This item is in the early stages; work will continue on the item working toward materials to aid in the training of field inspectors. It was indicated that working in conjunction with the Professional Development Committee (PDC) to develop training materials, etc. would be a logical path of progress once we have developed the information content to include.

At the 2011 NTETC Software Sector Meeting, it was suggested that this topic should be tabled for the time being, until items 1 – 4 in the Summary are better defined. This will also depend on the needs of and feedback from field inspectors, since the goal is to empower them to be better able to handle inspection of software-based devices.

Conclusion:
This item will be tabled until the next meeting. The Sector chair will liaise with the PDC to garner input for focus areas related to inspecting software-based devices where additional materials would be most beneficial to the needs of field inspectors.

NEW ITEMS

7. Remote or Distributed Metrologically Significant Functionality

Source:
California NTEP Laboratory

Background / Discussion:
A database on a remote server contains metrological data for a commercial transaction. The server storage containing the database is leased and access is granted for analysis manipulation, viewing, and/or printing the transaction data.

Previously the sector has discussed situations where data that is used as part of a transaction (e.g. tare values) are being retrieved from a remote server, but examples can be given that extends the boundaries. Is it acceptable to allow situations where data printed on the transaction report is not locally available or cannot be reproduced without server access? What about situations where actual metrologically significant software routines are executed on a remote server? Does this relate to what the WELMEC working group on software terms "Data Transmission"?

The following questions were raised during the initial discussion of this item to clarify the issue:

Questions:
- What happens if communication fails?
- How will printing be performed at the local site?
- Is it possible to print at the local site under those circumstances?

Answers:
- The printer was at the local site, but all of the information sent to the printer came from the remote site. It wasn't known if/how printing would work in case of communications failure.

The opinion was that the particular situation described wouldn't be a violation of *NIST Handbook 44*, so long as the first indication of final quantity is local.

It was stated that the only way to seal a system like this is via a Category 3 audit trail. Type approval should verify the accuracy and integrity of the communication between the remote component(s) and the local component(s). The factor that concerns the field inspectors the most is that the metrological calculations are being performed remotely. The field inspector can ask which factors are used in the metrological calculations and verify the output.

OIML D31.5.2.3 is relevant to this discussion:

> "5.2.3 Storage of data, transmission via communication systems. If measurement values are used at another place than the place of measurement or at a later time than the time of measurement they possibly have to leave the measuring instrument (electronic device, subassembly) and be stored or transmitted in an insecure environment before they are used for legal purposes."

Also relevant is WELMEC 7.2 Issue 5: 7.2 Specific Software Requirements for Data Transmission.

Of course these references shouldn't be assumed to be taken verbatim for NCWM purposes, but they can be used for guidance in the sort of questions that should be asked, during type approval and with the checklist used by the labs, as well as potentially by field inspectors.

Conclusion:No action. This must be resolved by the state jurisdiction.

8. Next Meeting

Background:
The sector is on a yearly schedule for NTETC Software Sector Meetings. Mr. Truex, NTEP Administrator, will determine when the next meeting is possible. The normal rotation would have the meeting in Columbus in 2012.

Background / Discussion:
none

Conclusion:Mr. Truex, NTEP Administrator, will arrange with the cooperation of the state of Ohio to host the next meeting in Columbus in 2012, probably in mid-March.

9. Report on 2011 Interim Meeting

There was one item on NCWM S&T Committee Agenda for the 2011 NCWM Interim Meeting related to work done by the sector. *2011 NCWM Publication 15* S&T Item 310-2 relates to the sector 2011 Agenda Item 1 (Marking Requirements). After some discussion, mostly supportive but tentative, the Chair had the impression that the bulk of the feedback seemed to indicate that the goals of the proposal are worthwhile but the language is still not satisfactory or sufficiently clear to some.

Report from NIST and other attendees of NCWM Interim Meeting:
The recommendation on Identification of Software was for it to remain Informational. SMA was the most reluctant to adopt the differentiation in types of software. Their feedback is based upon the idea that all types of software should have the same marking options.

10. Report on International Weights and Measures Activity

Highlights of interest to the NTETC Software Sector:

- CIML meeting in Orlando, Florida
- MAA updates
- Steve Patoray appointed International Bureau of Legal Metrology (BIML) Director last September. Took the position in January.
- New draft WELMEC 7.2 circulated in February of this year for comment.
- Workshop on Operating Systems in Legal Metrology hosted by PTB December 2010
- The second OIML document, for verification, was to be generated, but Germany doesn't seem to be working on it. The US (and Canada) have the opportunity to drive this development.
- PTB held a workshop in Berlin in December regarding OS's and legal metrology.
- The director of BIML has and the president of International Committee of Legal Metrology will be changing personnel.
- OIML D11 will be having a meeting in June.

- MAA Updates: There was a special vote to determine whether to accept manufacturers' test data. This was voted down.
- There's a new WELMEC 7-2 v.5 draft including mainly editing changes but also new information regarding operating systems.

ATTENDANCE

Dennis Beattie
Measurement Canada
400 St. Mary Ave
Winnipeg, MB R3C 4K5
Canada
(204) 983-8910
dennis.beattie@ic.gc.ca

Doug Bliss
Mettler-Toledo, Inc.
1150 Dearborn Drive
Worthington, OH 43085
(614) 438-4307
doug.bliss@mt.com

Mike Frailer
Maryland Weights and Measures
50 Harry S Truman Parkway
Annapolis, MD 21401
(410) 841-5790
fraileml@mda.state.md.us

Andy Gell
Foss North America
8091 Wallace Road
Eden Prairie, MN 55344
(952)974-9892
agell@fossnorthamerica.com

Teri Gulke
Liquid Controls, LLC
105 Albrecht Drive
Lake Bluff, IL 60044
(847) 283-8346
tgulke@idexcorp.com

Jody Hirst
Itron, Inc.
1310 Emerald Road
Greenwood, SC 29646
(864) 942-2245
jody.hirst@itron.com

Norman Ingram
California Division of Measurement Standards
6790 Florin Perkins Road
Suite 100
Sacramento, CA 95828
(916) 229-3016
ningram@cdfa.ca.gov

Mike Kelley
Ohio Department of Agriculture
8995 East Main Street
Reynoldsburg, OH 43068
(614) 728-6290
mkelley@agri.ohio.gov

Paul A. Lewis, Sr.
Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868
(715) 434-5322
plewis@ricelake.com

Rick Lydon
Sick, Inc.
800 Technology Center Drive
Suite 6
Stoughton, MA 02072
(781) 302-2552
richard.lydon@sick.com

Mike McGhee
Itron, Inc.
1310 Emerald Road
Greenwood, SC 29646
(864) 223-1212
michael.mcghee@itron.com

Ed Payne
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
(410) 841-5790
payneea@mda.state.md.us

Jim Pettinato
FMC Technologies Measurement Solutions, Inc.
1602 Wagner Avenue
Erie, PA 16510
(814) 898-5250
jim.pettinato@fmcti.com

Dan Reiswig
California Division of Measurement Standards
6790 Florin Perkins Road
Sacramento, CA 95828
(916) 229-3023
dreiswig@cdfa.ca.gov

Chris Scott
Gilbarco, Inc.
7300 W Friendly Ave
Greensboro, NC 27420
(336) 547-5227
chris.scott@gilbarco.com

Scott Szurek
Emerson Process Management
301 South 1st Ave
Marshalltown, IA 50158
(641) 754-3425
scott.szurek@emerson.com

Ambler Thompson
NIST Weights and Measures Division
100 Bureau Drive
MS 2600
Gaithersburg, MD 21701
(301) 975-2333
ambler@nist.gov

James Truex
National Conference on Weights and Measures
88 Carryback Drive
Pataskala, OH43062
(740) 919-4350
jim.truex@ncwm.net

John Wind
Bizerba USA, Inc.
5200 Anthony Road
Sandston, VA 23150
(804) 221-9699
john.wind@bizerba.com

National Type Evaluation Technical Committee (NTETC) Software Sector Meeting Summary

March 20-21, 2012 / Columbus, Ohio

INTRODUCTION

The charge of the NTETC Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the National Type Evaluation Program (NTEP) Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be nonretroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content | **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
CC	Certificate of Conformance	OIML	International Organization of Legal Metrology
CRC	Cyclical Redundancy Check	OWM	Office of Weights and Measures
EPO	Examination Procedure Outline	PDC	Professional Development Committee
NCWM	National Conference on Weights and Measures	PTB	Physikalisch-Technische Bundesanstalt
NIST	National Institute of Standards and Technology	S&T	Specifications and Tolerances Committee
NTEP	National Type Evaluation Program	SMA	Scale Manufactures Association
NTETC	National Type Evaluation Technical Committee	WELMEC	European Cooperation in Legal Metrology

Details of All Items
(In order by Reference Key)

WELCOME / INTRODUCTIONS

Mr. Pettinato, Chair, would like to welcome new individuals that have joined the NTETC Software Sector since the last meeting. Please welcome:

- Ms. Mary Abens, Emerson Process Management
- Mr. Thomas Fink, ITW Food Equipment/Hobart
- Mr. Adam Oldham, Gilbarco, Inc.

STATUS REPORTS

1. 2012 NCWM Interim Meeting Report

Source:
NCWM S&T Committee Agenda

Background / Discussion:
There was one item on the NCWM S&T Committee Agenda for the 2012 NCWM Interim Meeting related to work done by the NTETC Software Sector. *2012 Publication 15* S&T Item 360-2 relates to the 2012 NTETC Software Sector Agenda Item 1: Marking Requirements.

Conclusion:
Attendees indicated that the 2012 Interim Meeting was well attended. Most issues were not S&T issues – more laws and packaging type issues. The one issue that was on the S&T Committee Agenda has been changed from Informational to Developing. Mr. Truex, NTEP Administrator, was not at the Open Hearings when that item was discussed, but Mr. Lewis, Rice Lake Weighing Systems, Inc. was. He said it didn’t go anywhere.

2. 2012 International Activity Report

Source:
NTETC Software Sector

Background / Discussion:
Dr. Thompson, National Institute of Standards and Technology (NIST), Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the sector. Mr. Pettinato, Chair, will summarize the discussion that took place at the European Cooperation in Legal Metrology (WELMEC) WG7 meeting in December 2011.

Conclusion:
Highlights of interest to the NTETC Software Sector:

- Workshop on Operating Systems in Legal Metrology hosted by Physikalisch-Technische Bundesanstalt (PTB) December 2011 coincident with WELMEC WG7 meeting.
- New D-11 draft circulated for comment early 2012.

Mr. Pettinato, Chair attended the WELMEC WG7 meeting in Berlin in December. He was struck by how similar the discussion was to our NCWM meetings. We are trailing in requirements for software security. They are trying to enforce authentication, identification, self-checking, etc. They're dealing with Linux and other open-source issues. Some approvals have taken 18 months. They seem to be starting in a new direction, possibly rewriting D-7.2 to reference software documents for IT standards for security. This would result in them only focusing on metrological issues in the software, leaving the other standards to cover the remaining issues in security. Currently PTB references a National Security Agency document on securing Red Hat Linux.

Mr. Beattie, Measurement Canada, asked about the feeling regarding Common Criteria. Mr. Pettinato reported that there were a couple presentations on this subject. There are big concerns about data privacy. PTB has backed off from this approach since they've realized that it puts a lot of responsibility on their plate. This is part of why they are looking to recommend various IT standards. Dr. Thompson reported that the Germans had wanted to go to the extreme of detailed code-walking. Mr. Oldham, Gilbarco, Inc., mentioned that though Europe has apparently backed off on this, India and Mexico appear to be continuing to pursue it.

CARRY-OVER ITEMS

3. Software Identification / Markings

Source:
NTETC Software Sector

Background / Discussion:
Since its inception the sector has wrestled with the issue of software identification and marking requirements. *See the 2011 Software Sector Meeting Summary and the 2012 Interim Meeting S&T Agenda Item 360-2 for more background on this item.*

NIST, OWM had been adding items to the S&T Agendas that confused matters since the perception was that this sector had contributed to this input. Most of the confusion arose in the 1990's, due to some items being approved, and others, such as the definitions for "Built for Purpose" and "Not Built for Purpose," not being approved.

Mr. Truex, NTEP Administrator, discussed the difficulty there has been in coming to a consensus on these issues with a representative of the NTEP Committee. Suggestions from NTEP to come to some resolution has been to write an article for the newsletter (which Mr. Bliss, Mettler-Toledo, LLC, had already done, to no effect), sending a questionnaire to the NTEP community, asking what they'd like to see, and sending a representative from this sector to the S&T Committee.

Mr. Roach, California Division of Measurement Standards, is concerned that some people may want to interpret G-S.1.c as requiring a serial number for software. Mr. Lewis, Rice Lake Weighing Systems, Inc. pointed out that the computer that the software was running on could have the serial number, not the software itself. That shouldn't matter, regardless.

Mr. Bliss, Mettler-Toledo, LLC, pointed out that the terminology in G-S.1. "All equipment", could be interpreted to mean that it doesn't apply to software. It was proposed that G-S.1.c be amended to add "and software". Mr. Bliss suggested submitting a document explaining the reasoning behind the proposed changes, rather than assume that the text is self-explanatory. Making a presentation to the various committees on the subject in addition would be beneficial as well. If a document is written, perhaps the examples given in G-S.1.d.3.a can be eliminated. "Metrologically significant" isn't explicitly defined, but it's been used since time immemorial.

Attempts to modify G-S.1.1. have been controversial, both in this meeting and in other committees. Unfortunately, there has been little constructive feedback from the other committees. It would probably be easier to incorporate specific examples given in G-S.1.1.b.3 in *NCWM Publication 14*. After some discussion, the previously proposed language was modified slightly to address some of the concerns received via feedback from other sectors and interested parties:

NIST Handbook 44 – Proposed changes:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts and ~~**not-built-for-purpose software-based software devices**~~ **software**;
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier ~~**for not-built-for-purpose software-based electronic devices**~~;
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:

(a) The user interface does not have any control capability to activate the indication of the version or revision identifier on the display, or the display does not technically allow the version or revision identifier to be shown (analog indicating device or electromechanical counter) or

(b) the device does not have an interface to communicate the version or revision identifier.

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC.

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~**and,**~~ 2006 **and 201X**)

***G-S.1.1. Location of Marking Information for ~~Not Built For Purpose~~ all Software-Based Devices.** —For ~~**not built for purpose,**~~ software-based devices, either:*

*(a) The required information in G-S.1. Identification. (a), (b), ~~**(d),**~~ and (e) shall be permanently marked or continuously displayed on the device; or*

(b) The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."

***Note:** For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

[Nonretroactive as of January 1, 2004]

(Added 2003) (Amended 2006 **and 20XX**)

The new language in G-S.1.1 reflects that the sector reached consensus on the following positions:

- The software version/revision should (with very few exceptions – see D-31 5.1.1) be accessible via the user interface.
- The means by which the software version is accessed must be described in the Certificate of Conformance (CC).

In addition, it was asserted that the previously recommended changes to G-S.1.1 (b)(3) in fact are not really necessary; the current language of *NIST Handbook 44* empowers the laboratories to enforce “easily recognizable” as they see fit. In fact, the previously generated “list” of icons and menu options could certainly be used by the examining laboratories as part of the approval process (e.g. in *NCWM Publication 14*). Of course, a manufacturer who is reviewing *NIST Handbook 44* so as to develop an acceptable device may benefit from more explicit guidance. Where does such guidance belong?

Comments related to the circulated list included a comment from the Scale Manufacturers Association (SMA) suggesting that a definition is needed for “software-based devices.” SMA opposed the definitions previously put forth by the sector. It was suggested that perhaps SMA would be more amenable to a definition that doesn’t differentiate between software types.

The conclusion from the 2011 NTETC Software Sector Meeting was that the sector will request feedback on the new recommended language for G-S.1 and G-S.1.1 since it does deviate somewhat from previous submissions. It is hoped that the various interested sectors, regions and associations will give this new proposal careful thought and submit their concerns to the NTETC Software Sector.

The list of suggested icons/menus that should be considered finite options for manufacturers was updated to reflect comments received by the sector. The sector now believes this approach is adequate without a change to *NIST Handbook 44*; the NTEP laboratories would be able to enforce “easily recognizable” against this finite list. Hence, the sector recommends the list be inserted into *NCWM Publication 14*.

Crafting a definition for “software based device” may be included as an item in a future agenda. Note the term “not built for purpose, software based device” is already used in *NIST Handbook 44*.

Some concerns seemed to stem from a lack of understanding of intent. It was suggested that a supplementary document could be written, explaining the intent of the “software based device” terminology.

Conclusion:
The sector wishes to continue promotion of this item, with the minor edits shown above included addressing some of the concerns of other interested parties. Since this is currently defined as a Developing Item, it cannot be moved to a Voting Item at the 2012 NCWM Annual Meeting; it will have to wait until 2013. In January of 2013, the decision will be made as to changing the status of this item. This sector will need to push to accomplish this. Developing a presentation and/or writing a supplementary document that would explain the intent behind the proposed changes to G-S.1 and G-S1.1 would most likely help in getting these changes passed. The annual meeting would be an appropriate venue for a presentation, though it may be too late to get it onto the agenda. The SMA is having their meeting next month in Monterey, California. Mr. Fink, ITW Food Equipment/Hobart, may be available to assist Mr. Pettinato, Chair, in putting together a presentation and volunteered to present it at the SMA Meeting.

4. Identification of Certified Software

Source:
NTETC Software Sector

Background / Discussion:
This item originated as an attempt to answer the question “How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?” In previous meetings it was shown that the international community has addressed this issue (both WELMEC and International Organization of Legal Metrology (OIML)).

From WELMEC 7.2:

> **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**.**

From OIML D-31:

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- Cyclical Redundancy Check (CRC)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the weights and measures inspector information that something has changed?
Yes, the Category III Audit Trail or other means of sealing.

How can the weights and measures inspector identify an NTEP Certified version?
They can't, without adding additional requirements such as those described here, in conjunction with including the identifier on the CC.

The sector has continued to believe that we should work towards language that would include a requirement similar to the OIML requirement in *NIST Handbook 44*. It is also the opinion of the sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

Closely related to this concept of uniquely identifying software is the practice of software separation. The sector sees the benefit in allowing that metrological software be separated from non-metrological software for ease of identification and evaluation.

From OIML:

> 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)

Previously recommended text intended to be added to *NCWM Publication 14* was discussed and modified slightly*:*

Identification of Certified Software:

> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~**The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.**~~
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

From OIML D-31:

> Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.
>
> The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.

The first sentence of the first paragraph above is already addressed in *NIST Handbook 44's* marking requirements.

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

> *(d) the current software version or revision identifier for ~~**not built-for-purpose**~~ **<u>software-based electronic</u>** devices;*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) **<u>(Amended 20XX)</u>**
>
> *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> *[Nonretroactive as of January 1, 2007]*
> *(Added 2006)*
>
> *(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).*
> *[Nonretroactive as of January 1, 2007]*
> (Added 2006)
>
> **<u>(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.</u>**
> ***<u>[Nonretroactive as of January 1, 201X]</u>***
> **<u>(Added 20XX)</u>**

Also the sector recommends the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc (crc32, for example)

There was some additional discussion on this item regarding where this new requirement was best located. It was suggested that the first sentence of G-S.1.d.(3) could be added as a clause to the base paragraph G-S.1(d) text, e.g.

> *"the current software version or revision identifier for* ***~~not built for purpose~~*** ***software-based*** *devices, which shall be directly and inseparably linked to the software itself;"* .

It also was suggested that the second sentence in G-S.1.d. (3) might be more suitable for *NCWM Publication 14,* as it describes more "how" than "what" the requirement entails.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions that are still outstanding:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to "inseparably link" the identifier to the software? If it's acceptable to hard-mark the version or revision, the requirement to inseparably link it to the software is waived.
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked? If the device is capable of doing so, it must.

At the 2012 NTETC Software Sector Meeting, there was some discussion as to where the terminology regarding inextricably linking the software version or revision to the software itself belonged. At the moment, it is not incorporated in the proposed text for G-S.1. *NCWM Publication 14* may be a better option for the time being. This would be another item that would benefit from further explanation in a supplementary document.

One suggestion was this revision to G-S.1.d:

(d) ~~when metrologically significant software is employed,~~ the current software version or revision identifier, which shall be directly and inseparably linked to the software itself;~~, for not-built-for-purpose software-based electronic devices;~~

Alternatively, if the previously proposed new subsection G-S.1.d.3 from Item 1 is adopted, this concept could be inserted thus:

> *(3) The version or revision identifier shall be* directly and inseparably linked to the software itself and *accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:*

Several sector members were of the opinion that attempting to make this change at the same time as the earlier changes might be a difficult sell. Mr. Truex, NTEP Administrator, reiterated the necessity of baby steps.

Conclusion:
The sector recommends adding the following to *NCWM Publication 14* and forward to NTETC Weighing, Measuring, Grain Analyzer sectors for feedback:

Identification of Certified Software:

> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

Also, it was decided to forward the two alternate options for adding requirements for uniquely identifying software to the individual sectors:

One suggestion was this revision to G-S.1.d:

> (d) ~~when metrologically significant software is employed,~~ the current software version or revision identifier, which shall be directly and inseparably linked to the software itself;~~, for not built-for-purpose software-based electronic devices;~~

Alternatively, if the previously proposed new subsection G-S.1.d.3 from Item 1 is adopted, this concept could be inserted thus:

> *(3)* *The version or revision identifier shall be* **directly and inseparably linked to the software itself and** *accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:*

Both alternatives will be sent to the sectors for feedback.

5. Software Protection / Security

Source:
NTETC Software Sector

Background / Discussion:
The sector agreed that *NIST 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 of development techniques have been applied.
>
> This requirement includes consideration of:
>
> 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 *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTETC Software Sector Meeting. 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 laboratories agreed to use this check list on one of the next devices they have in the laboratory and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

1. Devices with Embedded Software ~~TYPE P (aka~~ built-for-purpose~~)~~

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **AND Needs clarification** □ Yes □ No □ N/A

1.2. 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.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the ~~securing~~ means **of sealing** (evidence of an intervention). **(Note: See Philosophy of Sealing in Pub. 14.)** □ Yes □ No □ N/A

1.3.3. Software Identification □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. **Describe how the identification applies to the software – is the metrological software separated or does the identifier apply to the entire software?** □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

2. 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~~)~~

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. **This may be part of the standard documentation, or it may be a separate document.** □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. **Can someone overwrite it or modify it after it's been installed without any evidence of a change?** □ Yes □ No □ N/A

2.2. 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

3. Software with Closed Shell (no access to the operating system and/or programs possible for the user). Shell means command-line interface or access to the Windows Desktop, as examples. This doesn't guarantee that there is no back door, just that the manufacturer doesn't know of one.

3.1. 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

3.2. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User

4.1. 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 Weights and Measures jurisdiction and type-specific parameters). **Is there a means to determine that the software is complete and authorized by the vendor – not damaged or someone else's program?** □ Yes □ No □ N/A

4.2. 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). **If the software is altered, is there some means to determine whether that has occurred? As an example, can an average text editor cause damage?** □ Yes □ No □ N/A

5. Software Interface(s)

5.1. **This is intended to determine whether the manufacturer has at least considered these issues.** Verify the manufacturer has documented:

5.1.1. The program modules of the metrologically significant software are defined and separated. **Has the metrologically significant software been separated from the other software?** □ Yes □ No □ N/A

5.1.2. The protective software interface itself is part of the metrologically significant software. **This is something that's used to close access to the metrologically significant software.**

5.1.3. The functions of the metrologically significant software that can be accessed via the protective software interface. **This could be all, none, or some. Functions mean more than just changing parameters. As an example, this may mean whether you can take a tare or not.** □ Yes □ No □ N/A

5.1.4. The parameters that may be exchanged via the protective software interface are defined. **The sealed parameter list from Pub. 14.** □ Yes □ No □ N/A

5.1.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.1.6. There are software interface instructions for the third party (external) application programmer. **If so, how is the metrologically-significant data and functionality protected? What can it do? Is it fixed? Can it be expanded?** □ Yes □ No □ N/A

The Maryland laboratory had particular questions regarding 3.1 and 5.1. The information for 3.1 could be acquired from an operator's manual, a training video, or in-person training. The items in 5.1 were confusing to the evaluators. The terminology is familiar to software developers, but not necessarily others. It was indicated that manufacturers were typically quick to return the filled out questionnaire, but he didn't know how his laboratory was supposed to verify that it was true. Generally, the laboratories wouldn't be expected to verify things to that level. For example, if the manufacturer states that a checksum is used to ensure integrity, the laboratories wouldn't be expected to evaluate the algorithm used.

The intent was to see whether the manufacturer had at least considered these issues, not for evaluators to become software engineers. Perhaps a glossary or descriptive paragraphs might be added to assist the evaluators for if the manufacturer has questions for the evaluators.

OIML makes use of supplementary documents to explain the checklist they use. Below are links:

- http://www.oiml.org/publications/D/D031-e08.pdf
- http://www.welmec.org/latest/guides/72.html
- http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf

WELMEC document 2.3 is the original source for our checklist, but it's been significantly revised and simplified. Mr. Payne, Maryland Department of Agriculture, is going to review the other documents and come up with some suggestions for the checklist. Mr. Roach, California Division of Measurement Standards, is going to begin using the checklist. The international viewpoint is that any device running an operating system is considered to be Type U. Mr. Roach mentioned that they're having lots of problems with "skimmers" stealing PIN's. Is there some way they can detect this?

Mr. Lewis, Rice Lake Weighing Systems, Inc., mentioned that he liked Measurement Canada's website. When answering similar questions, different pages would appear, based on answers to those questions: http://www.ic.gc.ca/eic/site/mc-mc.nsf/eng/lm00573.html

At the 2011 NTETC Software Sector Meeting, the laboratories were polled to obtain any feedback on the use of the checklist. Maryland attempted to use this checklist a few times. They had some difficulty obtaining answers from the manufacturers because the individual(s) interacting with the Maryland evaluator didn't always have the required information on hand. More experience in using the checklist will help determine what needs to be revised.

It was suggested that the checklist could be sent to manufacturers for their feedback as well, with the stipulation that it is a completely voluntary exercise and purely informational at this point. The laboratories will coordinate with willing manufacturers to obtain feedback.

Conclusion:
Work is ongoing on this item with the intent that it eventually will be incorporated as a checklist in *NCWM Publication 14*; again the laboratories are requested to try utilizing this checklist for any evaluations on software-based electronic devices.

The checklist has been reviewed with an eye to making its terminology clearer to laboratories. Some examples and clarifications have been added as shown in the discussion section of this item. The revised checklist will be distributed to the laboratories for additional review. Maryland and California laboratories agreed to use the checklist on a trial basis.

6. Software Maintenance and Reconfiguration

Source:
NTETC Software Sector

Background / Discussion:
After the software is completed, what do the manufacturers use to secure their software? The following items were reviewed by the sector.

1. Verify that the update process is documented.
2. For updates to be automatically verified by the device, installed software must be 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 such as 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 **<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 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 **<u>or become inoperative</u>**.

3. Verify that the sealing requirements are met
   This item is **<u>only</u>** addressing the **<u>software update</u>**, it can be either verified or traced.

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, i.e.

   "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 sector discussed how to best move this item forward, and there was also some discussion as to whether new language for the General Code was required. The general consensus of the group after considering feedback from external interested parties is that a new G-S.9 with explicit requirements is not necessary (nor likely to be adopted by NCWM) and that this requirement belongs in the *NCWM Publication 14* lists of sealable parameters rather than in *NIST Handbook 44*.

Additional work done at the 2011 NTETC Software Sector Meeting to further develop the proposed text toward inclusion in *NCWM Publication 14* resulted in the following:

> **<u>The updating of metrologically significant software shall be considered a sealable event. The software that checks for authenticity and integrity for a Traced Update, as well as the software responsible for generating and viewing the audit trail, is metrologically significant.</u>**
>
> **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**~~ Category 3 audit trail. The audit trail entry shall include the software identification of the newly installed version.

Note: It's possible that the Philosophy of Sealing section of NCWM Publication 14 may already address the above IF the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit:

Conclusion:

As a first step, attempt to add only the following to the Philosophy of Sealing in *NCWM Publication 14*:

> **The updating of metrologically significant software, including software that checks the authenticity and integrity of the updates, shall be considered a sealable event.**

Mr. Truex, NTEP Administrator, believes the above sentence is unnecessary since it's self-evident. It was agreed to ask the sectors for feedback on the value of this addition.

Though the sector is currently recommending only the single sentence be incorporated into *NCWM Publication 14* for the time being, ultimately, the sector may wish to advance the remaining language of the original item submission.

7. NTEP Application for Software and Software-based Devices

Source:

NTETC Software Sector

Background / Discussion:

The purpose of initiating this item was to identify issues, requirements and processes for type approving software applications. It was suggested that it may be useful to the laboratories to devise a separate submission form for software and devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, "The approval applicant is responsible for the provision of all the required equipment and components." This would likely also be the policy of NTEP.

Mr. Truex, NTEP Administrator, clarified that the current applications have some checks of software already, not that they couldn't benefit eventually from some additions. Once the checklist has been refined, it might prove useful. This won’t likely be a separate software checklist but rather some additional questions that will pertain to software, added to the existing list of questions that are currently asked.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4):

- A description of the software functions that are metrologically significant, meaning of the data, etc.
- A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).
- A description of the user interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.

Conclusion:
These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the laboratories is gathered.

8. Training of Field Inspectors

Source:
NTETC Software Sector

Background / Discussion:
During discussions at the 2009 NTETC Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *NIST Handbook 112** as a pattern template for how it could read.

Items to be addressed:

- CC
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources
- Safety

System Verification Tests

NOTE: Item numbers one through five apply to both weighing and measuring devices. Numbers six and seven are specific to weighing devices; while numbers nine and ten apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2
   Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.
   7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.
8. Over capacity.
   8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]
   8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.
   9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)
10. Over capacity.
    10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

This item is in the early stages; work will continue on the item working toward materials to aid in the training of field inspectors. It was indicated that working in conjunction with the Professional Development Committee (PDC) to develop training materials, etc. would be a logical path of progress once we have developed the information content to include.

At the 2011 NTETC Software Sector Meeting, it was decided that this topic should be tabled until items 1 – 4 in the summary are better defined. This will also depend on the needs of and feedback from field inspectors, since the goal is to empower them to be better able to handle inspection of software-based devices. It was also suggested that we liaise with the PDC to garner input for focus areas related to the inspection of software-based devices. It was also noted that OIML D-31 has sections on conformance to original type approval, etc., pertaining to software.

Conclusion:
Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices.*

NEW ITEMS

9. Next Meeting

Source:
NTETC Software Sector

Background / Discussion:
The sector is on a yearly schedule for NTETC Software Sector Meetings. Mr. Truex, NTEP Administrator, will determine when the next meeting is possible. The normal rotation would have the meeting in Sacramento, California in 2013.

Mr. Truex, NTEP Administrator, indicated that New York has re-established their laboratory, and would be an alternative site for the 2013 meeting. It was also mentioned that Sacramento had the benefit of Mr. Jordan, California Division of Measurement Standards, and/or Mr. Parks, California Division of Measurement Standards, being able to attend.

Conclusion:
The next meeting will be held either in Albany, New York or Sacramento, California depending on New York's willingness to host and locate an acceptable meeting location. Mr. Truex, NTEP Administrator, will make the determination as to meeting location by the end of the year.

10. NCWM Publication 14 Proposed Changes

Source:
NTEP Administrator

Background / Discussion:
Mr. Truex, NTEP Administrator, sent the sector membership a document outlining proposed changes to *NCWM Publication 14's Administrative Policy* section. The purpose is not to change the intent but to clarify it. He's asking for feedback on the proposed changes. Is the formatting, verbage, etc. acceptable? Does anyone have any questions or concerns? If so, send them to Mr. Truex. After the sectors have reviewed it, NTEP will do so, and then it will go to the Board of Directors.

Conclusion:
Members are asked to review and comment on the document and provide any feedback to the Mr. Truex, NTEP Administrator.

ATTENDANCE

Dennis Beattie
Measurement Canada
400 St. Mary Ave
Winnipeg, MB R3C 4K5
Canada
(204) 983-8910
dennis.beattie@ic.gc.ca

Doug Bliss
Mettler-Toledo, Inc.
1150 Dearborn Drive
Worthington, OH 43085
(614) 438-4307
doug.bliss@mt.com

Cassie Eigenmann
DICKEY-john Corporation
5200 Dickey-john Road
Auburn, IL 62615
(217) 438-2294
ceigenmann@dickey-john.com

Tom Fink
Hobart Corporation
701 Ridge Ave
Troy, OH 45374
(937) 332-3114
tom.fink@hobartcorp.com

Mike Frailer
Maryland Weights and Measures
50 Harry S Truman Parkway
Annapolis, MD 21401
(410) 841-5790
fraileml@mda.state.md.us

Teri Gulke
Liquid Controls, LLC
105 Albrecht Drive
Lake Bluff, IL 60044
(847) 283-8346
tgulke@idexcorp.com

Ken Jones
California Division of Measurement Standards
6790 Florin Perkins Road
Suite 100
Sacramento, CA 95828
(916) 229-3052
ken.jones@cdfa.ca.gov

Tom Junkans
Rice Lake Weighing Systems
230 West Coleman Street
Rice Lake, WI 54868
(715) 434-5130
tjunkans@ricelake.com

Paul A. Lewis, Sr.
Rice Lake Weighing Systems, Inc.
230 West Coleman Street
Rice Lake, WI 54868
(715) 434-5322
plewis@ricelake.com

Rick Lydon
Sick, Inc.
800 Technology Center Drive
Stoughton, MA 02072
(781) 302-2552
richard.lydon@sick.com

Paul McElhinney
Sick, Inc.
800 Technology Center Drive
Suite 6
Stoughton, MA 02072
(781) 302-2503
paul.mcelhinney@sick.com

Adam Oldham
Gilbarco, Inc.
7300 West Friendly Ave
Greensboro, NC 27420
(336) 547-5952
adam.oldham@gilbarco.com

Ed Payne
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
(410) 841-5790
payneea@mda.state.md.us

Jim Pettinato
FMC Technologies Measurement Solutions, Inc.
1602 Wagner Avenue
Erie, PA 16510
(814) 898-5250
jim.pettinato@fmcti.com

John Roach
California Division of Measurement Standards
6790 Florin Perkins Road
Suite 100
Sacramento, CA 95828
(916) 229-3456
john.roach@cdfa.ca.gov

Ambler Thompson
NIST, Office of Weights and Measures
100 Bureau Drive
Mail Stop 2600
Gaithersburg, MD 20899
(301) 975-2333
ambler@nist.gov

Jim Truex
National Conference on Weights and Measures
88 Carryback Drive
Pataskala, OH43062
(740) 919-4350
jim.truex@ncwm.net

John Wind
Bizerba USA, Inc.
5200 Anthony Road
Sandston, VA 23150
(804) 222-3922
john.wind@bizerba.com

National Type Evaluation Program (NTEP)
Software Sector Meeting Summary

March 19-20, 2013 / Columbus, Ohio

INTRODUCTION

The charge of the National Type Evaluation Program (NTEP) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the NTEP Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be nonretroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OIML	International Organization of Legal Metrology
CC	Certificate of Conformance	OWM	Office of Weights and Measures
EPO	Examination Procedure Outline	PDC	Professional Development Committee
GMMs	Grain Moisture Meters	PDC	Professional Development Committee
NCWM	National Conference on Weights and Measures	S&T	Specifications and Tolerances Committee
NTEP	National Type Evaluation Program	SMA	Scale Manufactures Association
NTETC	National Type Evaluation Technical Committee	WELMEC	European Cooperation in Legal Metrology

WELCOME / INTRODUCTIONS

The Chair would like to welcome new individuals that have joined the NTETC Software Sector since the last meeting. Please welcome:

- Eric Morabito, New York Bureau of Weights & Measures
- Gary Benjamin, NCR Corporation

STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

Attendees of the 2013 NCWM Interim Meeting will be asked to share any relevant comments or discussion that took place during the open hearings or NCWM Standards and Tolerances (S&T) committee working sessions.

Jim Truex was the only sector attendee at the interim meeting. He doesn't recall any comments on the floor. After the hearings, he had a brief discussion with the S&T committee, to little effect.

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the sector.

The new proposed revision of OIML has increased the risk classifications. The next CIML meeting is set for October.

CARRY-OVER ITEMS

1. Software Identification / Markings

Source:
NTETC Software Sector

Background:
Since its inception the sector has wrestled with the issue of software identification and marking requirements. *See the 2012 Software Sector Meeting Summary and the 2013 Interim Meeting S&T Agenda Item 360-2 for more background on this item.*

NIST,OWM had been adding items to the S&T Agendas that confused matters since the perception was that this sector had contributed to this input. Most of the confusion arose in the 1990's, due to some items being approved, and others, such as the definitions for "Built for Purpose" and "Not Built for Purpose," not being approved.

Mr. Truex, NTEP Administrator, discussed the difficulty there has been in coming to a consensus on these issues with a representative of the NTEP Committee. Suggestions from NTEP to come to some resolution has been to write an article for the newsletter (which Mr. Bliss, Mettler-Toledo, LLC, had already done, to no effect), sending a questionnaire to the NTEP community, asking what they'd like to see, and sending a representative from this sector to the S&T Committee.

Mr. Roach, California Division of Measurement Standards, is concerned that some people may want to interpret G-S.1.c as requiring a serial number for software. Mr. Lewis, Rice Lake Weighing Systems, Inc. pointed out that the computer that the software was running on could have the serial number, not the software itself. That shouldn't matter, regardless.

Mr. Bliss, Mettler-Toledo, LLC, pointed out that the terminology in G-S.1. "All equipment", could be interpreted to mean that it doesn't apply to software. It was proposed that G-S.1.c be amended to add "and software". Mr. Bliss suggested submitting a document explaining the reasoning behind the proposed changes, rather than assume that the text is self-explanatory. Making a presentation to the various committees on the subject in addition would be beneficial as well. If a document is written, perhaps the examples given in G-S.1.d.3.a can be eliminated. "Metrologically significant" isn't explicitly defined, but it's been used since time immemorial.

Attempts to modify G-S.1.1. have been controversial, both in this meeting and in other committees. Unfortunately, there has been little constructive feedback from the other committees. It would probably be easier to incorporate specific examples given in G-S.1.1.b.3 in *NCWM Publication 14*. After some discussion, the previously proposed language was modified slightly to address some of the concerns received via feedback from other sectors and interested parties:

NIST Handbook 44 – Proposed changes:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts and ~~**not built-for-purpose software-based software devices**~~ **software**;
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier ~~**for not-built-for-purpose software-based electronic devices**~~, ***which shall be directly linked to the software itself***;
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:

(a) The user interface does not have any control capability to activate the indication of the version or revision identifier on the display, or the display does not technically allow the version or revision identifier to be shown (analog indicating device or electromechanical counter) or

(b) the device does not have an interface to communicate the version or revision identifier.

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC.

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.) [Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~**and,**~~ 2006 **and 201X**)

G-S.1.1. Location of Marking Information for ~~Not Built For Purpose~~ all Software-Based Devices. *—For* ~~***not built for purpose,***~~ *software-based devices, either:*

(a) The required information in G-S.1. Identification. (a), (b), ~~***(d),***~~ *and (e) shall be permanently marked or continuously displayed on the device; or*

(b) The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."

Note: *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

[Nonretroactive as of January 1, 2004]

(Added 2003) (Amended 2006 **and 20XX**)

The new language in G-S.1.1 reflects that the sector reached consensus on the following positions:

- The software version/revision should (with very few exceptions – see D-31 5.1.1) be accessible via the user interface.
- The means by which the software version is accessed must be described in the Certificate of Conformance (CC).

The sector promoted this item following the meeting via several means to try and address the concerns of other interested parties. A presentation was generated and shared with the S.M.A. at their meeting. The regions had access to this information, as it was posted on the NCWM website. Unfortunately, based on the comments in the 2013 Pub 15 item 360-2, some regions were not aware that this information had been provided.

During the 2013 NCWM Interim Meeting, no comments were received relative to this item during the Open Hearings. In considering the item, the Committee questioned whether or not the Software Sector was still actively working the item. It was reported that the Software Sector believed they had developed the item as much as possible, yet the different stakeholders affected by the proposal could not agree on the changes that the Sector had proposed. Based upon that update, the Committee agreed to add to its report a request that the Software Sector work with the Weighing Sector and Measuring Sector to identify which portions of the proposal need to be modified in order that they might be accepted by the entire community. The Committee acknowledges and appreciates the efforts of the Software Sector and looks forward to being able to consider a proposal that addresses both the identification of software and how it may be accessed.

Discussion:
Since the 2012 meeting, the Sector has attempted to promote this item via several means to try and address the concerns of other interested parties. A presentation was generated and shared with the S.M.A. at their 2012 meeting. Most of the regions had access to this information prior to their meetings, as it was posted on the NCWM website. Unfortunately, based on the comments in the 2013 Pub 15 item 360-2, some regions were not aware that this information had been made available.

In addition, it was noted that it may be desirable to evaluate options that would lead to fully eliminating GS-1.1. It was noted that this would be a more invasive modification to the existing Handbook and perhaps should be put off until the first step of addressing software in all devices (not just standalone) was accomplished.

Conclusion:
The Sector considers this item sufficiently developed. The one response to our request for review/comment that contained negative feedback was undeniably vague and non-constructive. The issue seems to be more one of communication/understanding than disagreement with the intent or wording. We may want to consider more direct methods, i.e. designating a representative to address the regional groups or other sectors at their meetings. The annual meeting may be an appropriate venue for a presentation.

To move this forward, someone should address the regional groups. There are 5 – 6 potential venues for presentations. The last slide from the current presentation should be eliminated, to avoid confusing matters, for the time being. The two regional meetings in the fall (Western and Southern) and the interim meeting are probably more critical than the ones in May. Dr. Thompson was asked to relay that we have a presentation available and would like to push our proposal as a voting item in 2014. To be part of the January 2014 Annual S&T committee’s hearings / agenda, this needs to be brought to Rick Harshman’s attention. Dr. Thompson volunteered to speak with him.

After removing the “and inseparably” terminology from the proposal, the concerns on the possibility of controversy were reduced.

The sector’s opinion on the interpretation of “directly linked” is that it means that you can’t change the version/revision without changing the software.

It was recommended that a couple examples be added to the current slide presentation, to illustrate the intent of the proposed changes. One example might be supermarket-specific software designed to run upon a cash register. Another example might be, after a software change, noting that the new software version/revision number is no longer the same, and the operator was not prompted to enter a version/revision number.

Note: the text in red is a modification in the proposal made this year – the new text was inserted to address our Agenda Item 2. Upon the suggestion of NIST WMD, the modifications to Handbook 44 in these items were combined to avoid having to forward another proposal to modify Handbook 44 simultaneously or in the immediate future.

2. Identification of Certified Software

Source:
NTETC Software Sector

Background:

This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?" In previous meetings it was shown that the international community has addressed this issue (both WELMEC and OIML).

From WELMEC 7.2:

> **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**.**

From OIML D-31:

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the weights and measures inspector information that something has changed?
Yes, the Category III Audit Trail or other means of sealing.

How can the weights and measures inspector identify an NTEP Certified version?
They can't, without adding additional requirements like what is described here, in conjunction with including the identifier on the CC).

The sector believes that we should work towards language that would include a requirement similar to the International Organization of Legal Metrology (OIML) requirement in *NIST Handbook 44*. It is also the opinion of the sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

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

From OIML:

> 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)

Initial draft proposed language: (G-S.1.1?)

NIST Handbook 44 (This has been written into G-S.1.d.3): Identification of Certified Software:

> **Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified by the version or revision number**~~**.**~~ ~~**The identification**~~**, and this identification** ~~**of the software**~~ **shall be** ~~**inextricably**~~ **directly and inseparably linked to the software itself. The version or revision number may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**

From NCWM Publication 14:

Identification of Certified Software:

> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~**The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.**~~
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

From OIML D-31:

> Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.
>
> The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.

The first sentence of the first paragraph above is already addressed in *NIST Handbook 44's* marking requirements.

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

> *(d) the current software version or revision identifier for* ~~***not built for purpose***~~ ***software-based electronic*** *devices;*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) **(Amended 20XX)**
>
> > *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> > *[Nonretroactive as of January 1, 2007]*
> > *(Added 2006)*
> >
> > *(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin*

with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Also the sector recommended the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc (crc32, for example)

There was some additional discussion on this item regarding where this new requirement was best located. It was suggested that the first sentence of G-S.1.d.(3) could be added as a clause to the base paragraph G-S.1(d) text, e.g. "*the current software version or revision identifier for* ~~***not built for purpose***~~ ***software-based*** *devices, which shall be directly and inseparably linked to the software itself;*" .

It also was suggested that the second sentence in G-S.1.d. (3) might be more suitable for *NCWM Publication 14*, as it describes more "how" than "what" the requirement entails.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions that are still outstanding:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to "inseparably link" the identifier to the software?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

At the 2012 NTETC Software Sector Meeting, there was some discussion as to where the terminology regarding inextricably linking the software version or revision to the software itself belonged. At the moment, it is not incorporated in the proposed text for G-S.1. *NCWM Publication 14* may be a better option for the time being. This would be another item that would benefit from further explanation in a supplementary document.

Several sector members were of the opinion that attempting to make this change at the same time as the earlier changes might be a difficult sell. Mr. Truex, NTEP Administrator, reiterated the necessity of baby steps.

In 2012, the sector thus recommended adding the following to *NCWM Publication 14* and forward to NTETC Weighing, Measuring, Grain Analyzer sectors for feedback:

Identification of Certified Software:

> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

Discussion:
The Measuring Sector reviewed this item and had no feedback other than a statement that they support the continuing / ongoing efforts of this sector. The Weighing Sector summary mentioned that no one opted to provide comment. They agreed to take no further action on this item, pending further action from the Software Sector. This was specifically in reference to the accepted symbols.
For the time being, Jim Truex recommended that we not attempt to provide a definition for “software-based device”. We discussed the possibility of combining this change with the first agenda item, which had been attempted in previous years. Alternatively, if the HB44 changes from agenda item 1 are made, this agenda item could be addressed in Pub. 14.

Conclusion:
After further discussion, the wording in G-S.1.d under agenda item 1 was changed. Agenda item 2 will remain; however, it will address potential changes to Pub. 14 and contain no suggested modifications to Handbook 44. (See changes and conclusion under agenda item 1 for further details.)

The Sector chair volunteered to review the existing slide presentation detailing the purpose of these changes, to ensure that it accurately reflects this information.

The list of acceptable menu text and symbols in Appendix A are intended to assist the labs in finding the certification number. The sector noticed no action by the sectors had been taken when this list was circulated for comment. We would like to remind them that we would like to have it reviewed. We feel that this belongs in, for example, the Weighing Device Pub. 14, page DES-22, Section 3; the Belt – Conveyor Scales, page BCS-10, Section 8.7; the Measuring Devices, page LMD-21, Section 1.6; the Grain Moisture Meter, page GMM-14, Section 1 (G.S.1); and Near Infrared Grain Analyzers, page NIR-8, Section 1 (G.S.1).

3. Software Protection / Security

Source:
NTETC Software Sector

Background:
The sector agreed that *NIST 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 of development techniques have been applied.

This requirement includes consideration of:

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 *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTETC Software Sector Meeting. 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 laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

1. Devices with ~~Embedded~~ Software ~~TYPE P (aka built-for-purpose)~~

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **AND** □ Yes □ No □ N/A

1.2. 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.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, **including version / revision** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.1. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. □ Yes □ No □ N/A

2. ~~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)~~

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, **Cyclical Redundancy Check** (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. Software with ~~Closed Shell (~~no access to the operating system and/or programs possible for the user~~)~~

3.1. 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

3.2. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User

4.1. 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 Weights and Measures jurisdiction and type-specific parameters). □ Yes □ No □ N/A

4.2. Check whether the metrologically significant software will detect and act upon any unauthorized alteration of the metrologically significant □ Yes □ No □ N/A software using simple software tools (e.g., text editor).

5. Software Interface(s)

5.1. Verify the manufacturer has documented:

5.1.1. The program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.1.2. The protective software interface itself is part of the metrologically significant software. **□ Yes □ No □ N/A**

5.1.3. The functions of the metrologically significant software that can be accessed via the protective software interface. □ Yes □ No □ N/A

5.1.4. The parameters that may be exchanged via the protective software interface are defined. □ Yes □ No □ N/A

5.1.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.1.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

The Maryland laboratory had particular questions regarding 3.1 and 5.1. The information for 3.1 could be acquired from an operator's manual, a training video, or in-person training. The items in 5.1 were confusing to the evaluators. The terminology is familiar to software developers, but not necessarily others. It was indicated that manufacturers were typically quick to return the filled out questionnaire, but he didn't know how his laboratory was supposed to verify that it was true. Generally, the laboratories wouldn't be expected to verify things to that level. For example, if the manufacturer states that a checksum is used to ensure integrity, the laboratories wouldn't be expected to evaluate the algorithm used.

The intent was to see whether the manufacturer had at least considered these issues, not for evaluators to become software engineers. Perhaps a glossary or descriptive paragraphs might be added to assist the evaluators for if the manufacturer has questions for the evaluators.

OIML makes use of supplementary documents to explain the checklist they use. Below are links:
http://www.oiml.org/publications/D/D031-e08.pdf
http://www.welmec.org/latest/guides/72.html
http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf

WELMEC document 2.3 is the original source for our checklist, but it's been significantly revised and simplified. Mr. Payne, Maryland Department of Agriculture, is going to review the other documents and come up with some suggestions for the checklist. Mr. Roach, California Division of Measurement Standards, is going to begin using the checklist. The international viewpoint is that any device running an operating system is considered to be Type U. Mr. Roach mentioned that they're having lots of problems with "skimmers" stealing PIN's. Is there some way they can detect this?

Mr. Lewis, Rice Lake Weighing Systems, Inc., mentioned that he liked Measurement Canada's website. When answering similar questions, different pages would appear, based on answers to those questions: http://www.ic.gc.ca/eic/site/mc-mc.nsf/eng/lm00573.html

At the 2011 NTETC Software Sector Meeting, the laboratories were polled to obtain any feedback on the use of the checklist. Maryland attempted to use this checklist a few times. They had some difficulty obtaining answers from the manufacturers because the individual(s) interacting with the Maryland evaluator didn't always have the required information on hand. More experience in using the checklist will help determine what needs to be revised.

It was suggested that the checklist could be sent to manufacturers for their feedback as well, with the stipulation that it a completely voluntary exercise and purely informational at this point. The laboratories will coordinate with willing manufacturers to obtain feedback.

Work is ongoing on this item with the intent that it eventually will be incorporated as a checklist in *NCWM Publication 14*; again the laboratories are requested to try utilizing this checklist for any evaluations on software-based electronic devices.

The checklist has been reviewed with an eye to making its terminology clearer to laboratories. Some examples and clarifications have been added as shown in the discussion section of this item. The revised checklist will be distributed to the laboratories for additional review. Maryland and California laboratories agreed to use the checklist on a trial basis.

Discussion:
Over the past year, attempts to use the current checklist did not meet with many difficulties. The checklists were given to the manufacturers to fill out, and that seemed to work rather well. Minor modifications (in red above) were made to clarify certain confusing areas or eliminate redundancy.

Conclusion:
The next step will be to forward it to the four sectors; we can report that the labs have tried using it on a trial basis and we're ready to recommend it for Pub. 14 with the modification suggested here, such as the removal of the Type P / Type U wording.

4. Software Maintenance and Reconfiguration

Source:
NTETC Software Sector

Background:
After the software is completed, what do the manufacturers use to secure their software? The following items were reviewed by the sector. *Note that agenda Item 3 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 **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 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 weights and measures requirements.

The sector **agreed** that the two definitions below for Verified update and Traced update were acceptable.

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: It's possible that the Philosophy of Sealing section of NCWM Publication 14 may already address the above IF the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit).

Use of a Category 3 audit trail is required for a Traced Update. A log entry representing a traced software update shall include the software identification of the newly installed version.

The sector recommended consolidating the definitions with the above statement thus:

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**~~ Category 3 audit trail. The audit trail entry shall include the software identification of the newly installed version.

In 2012, the sector recommended that as a first step, the following be added to *NCWM Publication 14*:

The updating of metrologically significant software, including software that checks the authenticity and integrity of the updates, shall be considered a sealable event.

Mr. Truex, NTEP Administrator, indicated his opinion that the above sentence is unnecessary since it's self-evident. It was agreed by the group however to ask the other sectors for feedback on the value of this addition.

Though the sector is currently considering only the single sentence be incorporated into *NCWM Publication 14* for the time being, ultimately, the sector may wish to advance the remaining language of the original item submission.

Discussion:
The Sector had no information indicating that the other sectors had yet been approached for feedback on the value of the addition of the proposed sentence.

Conclusion:
This sector would like the other sectors to evaluate this for inclusion in Pub. 14.
We'd also like to include some description indicating that an existing audit trail should be protected during a software update, though that may already be a requirement. This does appear to be addressed in the Requirements for Metrological Audit Trails Appendices in Pub. 14.

5. NTEP Application for Software and Software-based Devices

Source:
NTETC Software Sector

Background/ Discussion:
The purpose of initiating this item was to identify issues, requirements and processes for type approving Type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, "The approval applicant is responsible for the provision of all the required equipment and components." This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4):

- A description of the software functions that are metrologically significant, meaning of the data, etc.
- A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).
- A description of the user interface, menus, and dialogs.
- A description of the method of sealing.
- The software identification (version, revision, etc.) and how to view it.

- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the ~~**operating system**~~ **software,** e.g. protection, user accounts, privileges, etc.
- The operating manual.

Conclusion:
The Sector recommends including the above bulleted list as an introduction to the checklist as part of our recommendation to include the checklist from agenda item 3 in Pub. 14. As a description of the accuracy of the measuring algorithms, simply declaring the type and class being aimed for may be sufficient. This list should reflect the needs of the labs for an evaluation. The bulleted list and the paragraph before it should be brought to the labs for an initial review and their input.

6. Training of Field Inspectors

Source:
NTETC Software Sector

Background:
During discussions at the 2009 NTETC Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources
- ~~**Safety**~~

System Verification Tests
NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).

4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.

5. Indications and displays.

   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.

   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4

   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4

7. Behind zero indication.

   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2

   Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.

   7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.

8. Over capacity.

   8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]

   8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.

   9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)

10. Over capacity.

    10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Co-Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices.*

Discussion:

The Sector would like to enlist field inspectors from a variety of states review California's Handbook 112, especially the excerpt above, to see if they think it would be of use to them. We'll obtain approval from California before we disseminate this documentation.

The PDC is focused on training sessions at the moment, so it's unsure how much time they'd have to review this currently.

There is a NIST/NCWM initiative on training. Dr. Thompson is going to bring this to their attention.

Aside from the general list of things to check, shown above, providing specific examples of scenarios they might encounter would likely be useful for field inspectors. A small working group, including Dr. Thompson, Ken Jones (or someone else from CA), Jim Pettinato, (check with Don Onweiler) possibly other field inspectors, etc. would be best to generate some examples.

A list of terms and acronyms could prove quite useful – not just to field inspectors, but perhaps even more so for type evaluators. The following is not really a list of definitions so much as various explanations of terms:

- CRC: Cyclical Redundancy Check
- Checksum
- Embedded software
- Firmware
- Version / Revision / Software Identifier: One component of a software identifier might be analogous to a model number, another component might be a version/revision, and another component might be a checksum. To satisfy the identification requirement, at a minimum, you need an identifier analogous to the model and a version / revision. In a product that has multiple pieces of software, you might require multiple software identifiers. For purposes of this list, Version and Revision are used synonymously.
- The difference between a serial number and a version / revision: Serial numbers are unique identifiers for a physical product. Identical copies of software can exist on multiple physical pieces of equipment, so serial numbers aren’t truly relevant to software. Instead, a version / revision number, tied to the software itself, is used to identify the differences between one set of software features and another. In summary, hardware needs a serial number, and software needs a version/revision number.
- Directly linked: Physical marking of hardware with a software version is useless as the software can be updated, in which case the physical marking would no longer be accurate. The preferred case is that the software self-identifies (displays version number, etc.) continuously or on demand. If the software changes, the version must change. There is an exception for situations where the device itself has no means to identify the software to the outside world, such as lacking a printer and a display.
- Hash: This is used for validation and verification that software and/or data is authentic and valid. A hash function is any algorithm or subroutine that maps large data sets of variable length to smaller data sets of a fixed length. Examples include CRC, checksum, LRC, etc. Hashes are used because there is a very low probability of two different data blocks having the same hash code.
- Signature
- Metrologically significant software: Software that calculates or affects features and/or measurements that are sealable.
- Software separation: Software can be divided into metrologically significant and non-metrologically significant sections. If it is, only metrologically significant software must be controlled. If separation is not employed, then the entire software is considered metrologically significant. “Controlled” implies that a separate software identifier for the metrologically significant software is used.
- Software update
- Sealable parameters: Reference Pub. 14. typical features or parameters to be sealed. Note that the download of software is recommended to be considered a sealable parameter.
- User interface: An interface forming the part of the instrument or measuring system that enables information to be passed between a human user and the measuring instrument or its hardware or software parts, such as, e.g. switch, keyboard, mouse, display, monitor, printer, touch-screen.
- Communications interface: An electronic, optical, radio, removable storage media, or other technical interface that enables information to be automatically passed between parts of measuring instruments, sub-assemblies, or external devices.
- Reset / reboot
- Non-volatile memory
- Flash
- Encryption

- Authentication: Affirmation that the source of the software or data was genuine and recognized. This can be done either via an authorized agent or via specific software techniques. Authentication is employed in order to prevent loading of malicious software into devices.
- Third party software: Software that is loaded into the weighing or measuring system that was not provided by the original manufacturer.
- Program
- Subroutine

We will flesh out this list, adding some brief definitions and/or examples. It will then be circulated amongst this group for review, and for any additional terms that are identified as being potentially useful.

Doug Bliss suggested developing educational presentations on relatively small software subjects, for presentation at the conferences, to provide training. We'll check into availability of time slots. January or next July are probably the earliest opportunities. Potential topics might include:

- General "software isn't scary"
- Background on why the software sector exists and what we're trying to accomplish
- Something to tie into the training of field inspectors on software
- Software identification
- Teaching inspectors how to read a certificate, with an eye toward information pertaining to software

Conclusion:
The Sector sees value in assisting in the training of field inspectors on several fronts as indicated by the discussion at this year's meeting. Several initiatives will be floated amongst the NCWM community and the Sector will focus on those that seem to have the most interest/benefit to the Conference.

NEW ITEMS

7. Next Meeting

Background:
The sector is on a yearly schedule for NTETC Software Sector Meetings. Mr. Truex, NTEP Administrator, will determine when the next meeting is possible. This year was California's turn in the rotation to host the meeting, but due to the uncertainly of New York's status as potential host, the meeting ended up being back in Ohio. Hence, New York and California again are possible locations for the 2014 meeting.

Albany, NY and California remain under consideration, with New York being the first choice, preferably as late as possible in March.

8. 2013 NCWM Interim Meeting Report

There was one item on the NCWM S&T Committee Agenda for the 2012 NCWM Interim Meeting related to work done by the NTETC Software Sector. *2012 Publication 15* S&T Item 360-2 relates to the 2012 NTETC Software Sector Agenda Item 1: Marking Requirements.

The Sector was informed of the S&T Committee decision to continue Item 360-2 as a Developing item.

9. 2013 International Report

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the sector. Software Sector Co-Chair Mr. Jim Pettinato will summarize the discussion that took place at the European Cooperation in Legal Metrology (WELMEC) WG7 meeting in Dec. 2011.

Highlights of interest to the NTETC Software Sector:

- New WELMEC 7.2 draft document circulated for comment by WG7
- R-117 working group

10. 360-7 D Appendix D – Definitions: Remote Configuration Capability

Source:
NTETC Grain Analyzer Sector (2013)

Purpose:
Expand the scope of definition to cover instances where the "other device," as noted in the current definition, may be necessary to the operation of the weighing or measuring device or which may be considered a permanent part of that device.

Item Under Consideration:
This item is under development. Comments and inquiries may be directed to NIST Office of Weights and Measures.

A proposal to modify the definition for "remote configuration capability" as follows is under consideration:

> **remote configuration capability. –** The ability to adjust a weighing or measuring device or change its sealable parameters from or through some other device that ~~**is not**~~ **<u>may or may not</u>** itself **<u>be</u>** necessary to the operation of the weighing or measuring device or ~~**is not**~~ **<u>may or may not be</u>** a permanent part of that device.[2.20, 2.21, 2.24, 3.30, 3.37, 5.56(a)]
>
> (Added 1993, **<u>Amended 20XX</u>**)

Background / Discussion:
Removable digital storage devices can be used in GMMs as either data transfer devices that are not necessary to the operation of the GMM or as data storage devices which are necessary to the operation of the GMM. If removal data storage devices are necessary to the operation of the device, they are not covered by the current definition of remote configuration capability.

A USB flash drive is most likely to be used as a data transfer device. In a typical data transfer application, the USB flash drive is first connected to a computer with access to the GMM manufacturer's web site to download the latest grain calibrations that are then stored in the USB flash drive. The USB flash drive is removed from the computer and plugged into a USB port on the GMM. The GMM is put into remote configuration mode to copy the new grain calibration data into the GMM's internal memory. When the GMM has been returned to normal operating (measuring) mode the USB flash drive can be removed from the GMM.

Although a Secure Digital (SD) memory card could also be used as a data transfer device it is more likely to be used as a data storage device. In a typical "data storage device" application, the SD memory card stores the grain calibrations used on the GMM. The SD memory card must be plugged into an SD memory card connector on a GMM circuit card for the GMM to operate in measuring mode. To install new grain calibrations the GMM must be turned "off" or put into a mode in which the SD memory card can be safely removed. The SD memory card can either be replaced with an SD memory card that has been programmed with the new grain calibrations or the original SD memory card can be re-programmed with the new grain calibrations in much the same way as that described in the preceding paragraph to copy new grain calibrations into a USB flash drive. In either case, the SD memory card containing the new calibrations must be installed in the GMM for the GMM to operate in measuring mode. In that regard, the SD memory card (although removable) can be considered a permanent part of the GMM in that the GMM cannot operate without it.

Note: In the above example SD memory card could be any removable flash memory card such as the Secure Digital Standard-Capacity, the Secure Digital High-Capacity, the Secure Digital Extended-Capacity, and the Secure Digital Input/Output, which combines input/output functions with data storage. These come in three form factors: the original size, the mini size, and the micro size. A Memory Stick is a removable flash memory card format, launched by Sony in 1998, and is also used in general to describe the whole family of Memory Sticks. In addition to the original Memory Stick, this family includes the Memory Stick PRO, the Memory Stick Duo, the Memory Stick PRO Duo, the Memory Stick Micro, and the Memory Stick PRO-HG.

At its 2011 Grain Analyzer Sector Meeting the sector agreed by consensus that the following changes to Table S.2.5. of §5.56.(a) of *NIST Handbook 44* should be forwarded to the S&T Committee for consideration:

- Add a note to Table S.2.5. to recognize the expanded scope of remote capability.
- Delete "remotely" from the second paragraph of Category 3 requirements that begins, "When accessed remotely …" to make it clear that the requirements of Category 3 apply whether accessed manually using the keyboard or accessed by remote means.
- Add the modified second paragraph of Category 3 requirements to Categories 3a and 3b to make it clear that these requirements apply to all the subcategories of Category 3.

After additional review of this item, the NIST, OWM recommended that the changes to Table S.2.5. approved by the sector in 2011 be separated into two independent proposals: one dealing with the changes to Category 3 and its subcategories and one recommending a modification of the definition of Remote Configuration Capability appearing in Appendix D of NIST Handbook 44 to recognize the expanded scope of remote capability, instead of adding a note to the bottom of Table S.2.5 to expanded the definition for remote configuration for grain moisture meters (as shown in this proposal). A change to the definition of remote configuration capability will apply to other device types.

2012 Grain Analyzer Sector Meeting: The sector agreed by consensus to separate its original proposal into two separate proposals and that this proposal to change the definition of Remote Configuration Capability should be forwarded to the S&T to Committee for consideration.

Item 5 of the NTETC, Grain Analyzer Sector August 2012 Meeting Summary covers this subject and will be available on NCWM Website November 2012.

2012 NCWM Annual Meeting: Ms. Juana Williams NIST, OWM supported the intent. She talked about this item in conjunction with Item 356-1: S.2.5. Categories of Device and Methods of Sealing. This is such a complex item affecting multiple other devices; therefore the proposal requires further consideration. The language in the proposal to amend the definition of remote configuration capability is confusing. The Committee believes the current definition already allows the use of remote configuration devices and allows the flexibility desired. The ramifications of changing the definition could affect other devices in HB 44. WWMA did not forward this item to NCWM.

2012 SWMA Annual Meeting: There were no comments. After reviewing the proposal and considering the potential impact on other device types, the Committee recommended this as a Developing Item. The Committee asks that the Sector continue to obtain input on the definition and the impact the changes would have on other device types. SWMA forwarded the item to NCWM, recommending it as a Developing Item and assigning its development to the Grain Analyzer Sector.

During its Open Hearings at the 2013 NCWM Interim Meeting, the Committee heard comments from Ms. Juana Williams (NIST OWM). OWM suggests the Committee consider this item as a Developing item to allow other

Sectors to discuss how a change to the definition may affect other device types of similar design and to consider changes if needed. OWM recognizes that the current definition for “remote configuration capability” may not address those grain moisture meters (GMMs) which can only be operated with a removable data storage device, containing, among other things, the grain calibrations intended for use with the GMM, inserted in the device (as was described by the Grain Analyzer Sector). As such, OWM notes that current sealing requirements were developed at a time when such technology likely didn’t exist, nor could be envisioned, and are based on the current definition of remote configuration capability. Because the current definition was never intended to apply to this “next generation” technology, OWM suggests that those charged with further development of this item may wish to revisit the five philosophies of sealing and consider whether a new paragraph, completely separate from current sealing requirements, might be appropriate and a better option, than the one currently proposed. The five philosophies of sealing are included in the 1992 Report of the 77th National Conference on Weights and Measures (Report of the Specifications and Tolerances Committee). Another option, preferred over the changes currently proposed, would be to add a separate statement to the current definition of “remote configuration capability” to address removable storage devices. For example, the following sentence might be considered as an addition to the current definition for “remote configuration capability:”

> **Devices which are programmed using removable media (such as SD cards, flash drives, etc.) that may or may not be required to remain with the device during normal operation are also considered to be remotely configured devices.**

The Committee also heard comments from Dmitri Karimov (LC), speaking on behalf of the MMA, who made two points: (1) Flow computers may already have these capabilities, thus it may be more appropriate to consider adding requirements to the General Code so that the requirements will be uniformly applied to all device types; and (2) the Committee should look ahead and consider other capabilities that may or already have emerged such as wireless communication and configuration.

The Committee acknowledged the comments indicating that the current definition of “remote configuration capability” was developed at a time when certain technologies, such as blue tooth, SD storage devices, flash drives, etc., didn’t exist. The Committee recognized that it may be difficult to modify the existing definition and associated requirements to be flexible enough to address emerging and future technologies without having a significant (and possibly detrimental impact) on existing devices. Consequently, rather than modifying the current definition, the Committee concluded that a better approach might be to develop an entirely separate set of security requirements that would apply to emerging technologies. The Committee believes that additional work is needed to develop proposed definition(s) and associated requirements and decided to designate the item as Developmental. The Committee requests other Sectors review the Grain Sector’s proposed modification to the definition as well as OWM’s suggestions and provide input.

Additional letters, presentations and data may have been part of the Committee’s consideration. Please refer to www.ncwm.net/content/2013pub-16 to review these documents.

Discussion:
Jim Pettinato and Doug Bliss suggested this alternative, possibly with the addition of some examples:

> **remote configuration capability. –** The ability to adjust a weighing or measuring device or change its sealable parameters from or through some other device ~~**that is not**~~ **may or may not** ~~**itself**~~ **be** ~~**necessary to the operation of the weighing or measuring device or is not**~~ **may or may not be** ~~**a permanent part of that device**~~.[2.20, 2.21, 2.24, 3.30, 3.37, 5.56(a)]
>
> (Added 1993, **Amended 20XX**)

This proposal is technology-agnostic and addresses the concern that any revision of the definition would be tied to existing technology.

The Sector is curious as to how updates to the calibration parameters via either USB or SD cards are being handled to date. For example, when replacing an SD card, are the parameter changes being recorded in an audit trail?

Conclusion:
We will forward this comment to the S&T Committee and the Grain Analyzer Sector.

A. Appendix – List of Acceptable Menu Text / Icons for Identification of Certificate Number

Table 1 - Software Sector Proposed Menu Text /Icons

Permitted Menu Text examples	*Permitted Icon shape examples*	*Essential characteristics*
Information Info	i i i	Top level menu text or icon • Icon text is a lower case "i" with block serifs • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Help ?	? ?	Top level menu text or icon • Icon text is a question mark • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Metrology Metrological Information	M M	Top or second level menu text or icon • Icon text is an upper case "M" • Text color may be light or dark but must contrast with the background color • Icon may have a circular, rectangular, or rounded rectangle border. • If present, the activation of this menu text/icon must recall at a minimum the NTEP CC number.
NTEP Data N.T.E.P. Certificate	NATIONAL CONFERENCE ON WEIGHTS AND MEASURES NTEP	This one is debatable – what if the certificate is revoked? Does NTEP grant holders of CCs the right to display the logo on the device, or just in documentation?
Weights & Measures Info	W&M W/M	

National Type Evaluation Program (NTEP)
Software Sector Meeting Summary

August 27-28, 2014 / Atlanta, GA

INTRODUCTION

The charge of the National Type Evaluation Program (NTEP) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the National Type Evaluation Program (NTEP) Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be nonretroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content | **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OWM	Office of Weights and Measures
CC	Certificate of Conformance	PDC	Professional Development Committee
EPO	Examination Procedure Outline	PDC	Professional Development Committee
GMMs	Grain Moisture Meters	S&T	Specifications and Tolerances Committee
NCWM	National Conference on Weights and Measures	SMA	Scale Manufactures Association
NTEP	National Type Evaluation Program	WELMEC	European Cooperation in Legal Metrology
OIML	International Organization of Legal Metrology		

Details of All Items

(In order by Reference Key)

WELCOME / INTRODUCTIONS

Since the first day of this year's Sector meeting was a joint meeting with the Weighing Sector, there was some time set aside to meet and greet both new and familiar faces. In addition, the Software Sector gave a brief presentation outlining the problems they've been asked to consider and some of the consensus that has been reached.

STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

Attendees of the 2014 NCWM Interim Meeting were asked to share any relevant comments or discussion that took place during the open hearings or NCWM Standards and Tolerances (S&T) committee working sessions.

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), provided a synopsis of international activity that relates to the work of the sector.

JOINT SESSION PROGRESS REPORT, ACTIVE ITEMS OF MUTUAL INTEREST

Since this is the first joint meeting of the Sectors, it is expected that some time will be required to review the agenda items of the Sectors that require collaboration, so all participants have a solid foundation for discussion. As part of this review, items of particular importance or interest should be allocated more time during the joint session day.

SOFTWARE SECTOR PRESENTATION

Doug Bliss, Software Sector technical advisor, gave a short presentation on the current issues being addressed by the Software Sector (see Appendix B) to the joint group. The presentation was well received; and generated some discussion. Adam Oldham pointed out that WELMEC doesn't go into minute detail regarding what is metrologically significant. He also asked how the manufacturer is intended to demonstrate the separation of software. Jim Pettinato responded that he thinks this will likely be a "paperwork demonstration", and that eventually we'll need to go into more detail on the subject. There was discussion of OIML's requirements, and how they're becoming less stringent over time.

Rainer Holmberg asked whether there have been problems with fraudulent software in the marketplace. Jim Truex said that instances have occurred in LA County and Detroit. There was also a problem with zero tracking that was found. Mike Wedman also related situations he'd encountered in the field that obviously did not provide sufficient protection of the software.

Jim Truex attempted to explain the direction we've been going in – though we are looking to OIML / WELMEC, there is no intent to go to their extent of detail.

The checklist that has been in development for inclusion in Publication 14 by the Software Sector was brought up during this discussion (see Agenda Item 3); it was pointed out that the Weighing Sector has already agreed to put the checklist into Pub. 14.

CARRY-OVER ITEMS

1. Software Identification / Markings

Sources:

- 2009 NTEP Software Sector Agenda Item 3 and 2010 S&T Item 310-3 G-S.1 Identification (Software)
- 2010 Final Report of the S&T Committee: ncwm.net/content/annual-archive
- 2010 Software Sector summary: http://www.ncwm.net/committees/ntep/sectors/software/archive
- 2011 Software Sector summary: http://www.ncwm.net/committees/ntep/sectors/software/archive
- 2011 Final Report of the S&T Committee (Publication 16 and addendum sheets): ncwm.net/content/annual-archive
- 2012 Software Sector summary: http://www.ncwm.net/committees/ntep/sectors/software/archive
- 2012 Final Report of the S&T Committee: http://www.ncwm.net/resources/dyn/files/1025938z8fff0401/_fn/2013_ST_Pub16.pdf
- 2013 Software Sector Summary: http://www.ncwm.net/resources/dyn/files/981560z45f7a5f5/_fn/12_Software_Sector_Activity.pdf
- 2013 Final Report of the S&T Committee: http://www.nist.gov/pml/wmd/pubs/sp1171.cfm
- 2014 Final Report of the S&T Committee: *To be added*

Background:

Local weights and measures inspectors need a means to determine whether equipment discovered in the field has been evaluated by NTEP. If so, the inspector needs to know at a minimum the CC number. From this starting point, other required information can be ascertained, e.g., the software version or revision identifier of the software installed in an electronic device at the time it was evaluated. NIST Handbook 44 currently includes three options for marking of the CC:

1. Permanent marking
2. Continuous display
3. Recall using a special operation

Additional background information relative to this item can be found in *2014 NCWM Publication 16* at: http://www.ncwm.net/resources/dyn/files/1217541z1019c056/_fn/4-ST-Pub16-2014-CORRECTED-06-12-2014.pdf

During its 2013 meeting, the WS, at the request of the SS, reviewed and provided feedback on the following SS proposal to amend NIST Handbook 44 General Code paragraphs G-S.1.Identification and G-S.1.1. Location of Marking Information for Not-Built-For-Purpose, Software-Based Devices:

NIST Handbook 44 – Proposed changes:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) a nonrepetitive serial number, except for equipment with no moving or electronic component parts and **~~not built-for-purpose software-based software devices~~** **software**;
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier **~~for not-built-for-purpose software-based electronic devices~~**, which shall be directly linked to the software itself;
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) ***The version or revision identifier shall be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:***

(a) ***The user interface does not have any control capability to activate the indication of the version or revision identifier on the display, or the display does not technically allow the version or revision identifier to be shown (analog indicating device or electromechanical counter) or***

(b) ***the device does not have an interface to communicate the version or revision identifier.***

(e) an NTEP CC number or a corresponding CC Addendum Number for devices that have a CC.

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device.

(Amended 1985, 1991, 1999, 2000, 2001, 2003, **~~and,~~** 2006 **and 201X**)

G-S.1.1. Location of Marking Information for ~~Not Built For Purpose~~ All Software-Based Devices. *– For* ***~~not built for-~~***

~~**purpose,**~~ *software-based devices, either:*

(a) The required information in G-S.1. Identification. (a), (b), ~~**(d)**~~**,** *and (e) shall be permanently marked or continuously displayed on the device; or*

(b) The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."

Note: *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*

[Nonretroactive as of January 1, 2004]

(Added 2003) (Amended 2006 **and 20XX**)

See the 2013 WS Final Report to view the feedback provided by the WS on the SS's proposal to amend paragraphs G-S.1. and G-S.1.1. and for additional background information relating to this item.

This item was also a "Developing" item on the 2014 S&T Committee's agenda and remains so on the 2015 S&T Committee's agenda. During the 2014 NCWM Annual Meeting, NIST OWM provided the following comments concerning the SS's proposal:

The following two concerns and suggestions were provided concerning the changes proposed to subparagraph G-S.1.(d):

1. Deleting the words "for not-built-for-purpose software-based electronic devices" creates the implication that all equipment manufactured as of January 1, 2004, except weights and separate parts necessary to the measurement process but not having any metrological effect, would be required to be permanently marked with a current software version or revision identifier. OWM questions whether or not it is the Software Sector's intent to require a software version or revision identifier be marked on equipment that is not electronic. If not the intent, OWM suggests that the Sector consider adding text to better clarify the type of equipment intended to be addressed by this proposed change and offers the following additional text for consideration:

(d) the current software version or revision identifier **for software-based electronic devices,** which shall be directly linked to the software itself;

2. The proposed changes, if adopted, would require a current software version or revision identifier be marked on both built-for-purpose and not-built-for purpose software based equipment manufactured as of January 1, 2004. If it is the intent of the Sector to require that a current software version or revision identifier be marked on built-for-purpose software based equipment, then the Sector might consider proposing that such a requirement be non-retroactive or that it become enforceable at some future date considering the time and cost involved in updating equipment already in service.

The following additional feedback was provided by OWM concerning the Software Sector's proposed changes to paragraphs G-S.1. and G-S.1.1.:

- It is not clear what equipment would be affected by the proposed changes to G-S.1. (c). By proposing that the word "software" be added, is the exception intended to apply to the software itself or to equipment in which the software is installed?
- In the proposed additions to G-S.1.(d)(3)(a), it is not clear what is meant by the phrase "or the display does not technically allow the version or revision identifier to be shown." The examples "analog indicating device" and "electromechanical counter" do not provide enough information to lead one to conclude that the intent is to address such things as numeric-only displays. That is, numeric-only displays that don't have the capability of displaying abbreviations for "version" or "revision" as noted in earlier comments originating from the Sector.
- OWM recommends adding some examples to clarify the types of devices described in paragraph G-S.1.(d)(3)(b).
- OWM agrees with the Software Sector's assertion that it may be possible to eventually eliminate G-S.1.1. at some future date.

OWM noted that a joint meeting of the Software and Weighing Sectors is planned in August 2014 to consider the current proposal and to try and reach agreement on the changes necessary to paragraph G-S.1. OWM encouraged the two sectors to consider its comments and feedback when considering any changes to the language currently proposed for G-S.1. The approach used in the past has been for the sectors to review the proposal in separate meeting sessions; however, this has not resulted in a proposal amenable to all sectors. OWM believes that it might be more expedient for all of the sectors to collaborate in a single joint meeting to try and reach agreement on the changes needed.

Following the 2014 NCWM Annual Meeting, members of OWM's Legal Metrology Devices Program (LMDP) were requested to provide additional input on the proposal to modify G-S.1. and G.S.1.1. in consideration of the goals of the SS and the comments provided during the 2014 Open Hearings of the S&T Committee relating to this item.

The following is a list of the goals provided by the SS in modifying G-S.1. and G.S.1.1. as communicated to the members of OWM's LMDP:

1. Remove the existing distinction between software identification requirements for built-for-purpose and not-built-for-purpose devices.
2. To require that <u>all</u> software-based devices have a software version or revision identifier for metrologically significant software.
3. Require that <u>certified</u> software versions or revision identifiers for metrologically significant software is recorded on the CC for access by inspectors.
4. Software itself does not require serial numbers.
5. Require that software-based devices version or revision identifier shall be accessible via the display and user interface and only if device's display is incapable of displaying the identifier or has no display and/or interface; then permanently marking the version or revision identifier shall be acceptable (e.g., digital load cell).
6. Nonretroactive as of January 1, 2016, if passed by the NCWM in July 2015.

OWM's LMDP developed the following proposed draft alternative changes to G-S.1. based on the SS's request for additional input on how best to meet its goals and forwarded them to the Chairman of the SS for consideration at the 2014 WS/SS joint meeting:

Amend NIST Handbook 44*:* G-S.1. Identification and G-S.1.1. Location of Marking Information for Not-Built-For-Purpose, Software-Based Devices as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.

[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and* ***~~not-built-for-purpose software-based devices~~*** ***software****;*

[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.

[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).

[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier for not-built-for-purpose software-based devices**;** **manufactured as of January 1, 2004 through December 31, 2015, and all software based devices or equipment manufactured as of January 1, 2016;**

~~[Nonretroactive as of January 1, 2004]~~
(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be:

i. prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;

[Nonretroactive as of January 1, 2007]
(Added 2006)

ii. directly linked to the software itself; and
[Nonretroactive as of January 1, 2016]
(Added 20XX)

iii. continuously displayed* or be accessible via the display menus. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable providing the device does not have an integral interface to communicate the version or revision identifier.

[Nonretroactive as of January 1, 2016] (Added 20XX)

****The version or revision identifier shall be displayed continuously on software-based equipment with a digital display manufactured as of January 1, 20XX and all software-based equipment with a digital display as of January 1, 20YY.***

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be

followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).
[Nonretroactive as of January 1, 2007]
(Added 2006)

(e) an National Type Evaluation Program (NTEP) Certificate of Conformance (CC) *number or a corresponding CC Addendum Number for devices that have a CC.*

<u>(1)</u> *The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)*
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device. (Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~**and,**~~ 2006 <u>**and 201X**</u>)

G-S.1.1. Location of Marking Information for ~~Not Built For Purpose~~ <u>All</u> Software-Based Devices. *– For ~~**not built for purpose,**~~ software-based devices, either:*

(a) The required information in G-S.1. Identification. (a), (b), ~~(d),~~ and (e) shall be permanently marked or continuously displayed on the device; or

(b) The CC Number shall be:

(1) permanently marked on the device;

(2) continuously displayed; or

(3) accessible through an easily recognized menu and, if necessary, a submenu. Examples of menu and submenu identification include, but are not limited to, "Help," "System Identification," "G-S.1. Identification," or "Weights and Measures Identification."

Note: *For (b), clear instructions for accessing the information required in G-S.1. (a), (b), and (d) shall be listed on the CC, including information necessary to identify that the software in the device is the same type that was evaluated.*
[Nonretroactive as of January 1, 2004]
(Added 2003) (Amended 2006 <u>**and 20XX**</u>)

No changes to subparagraph G-S.1.1 were proposed by OWM's LMDP since the SS had indicated earlier that it may be possible to eventually eliminate G-S.1.1. Thus, the proposed changes to subparagraph G-S.1.1 shown above in OWM's draft alternative changes are those originating from the SS's 2013 proposal.

In providing feedback to the SS, OWM's LMDP noted that the shaded portion of G-S.1(d)(1).iii of their draft alternative changes was developed solely by OWM (i.e., does not reflect any of the goals communicated by the SS) and was being offered for consideration with the understanding that:

1. this change will make it easier in the future for inspectors to be able to identify software installed in equipment;
2. a reasonable amount of time for the changes to take effect can be specified;

3. it is probable that improvements in technology over time will make it easier for equipment manufacturers to comply.

In addition to the alternative changes proposed by OWM's LMDP, a member of the SS submitted the following definition of "software-based devices" for discussion during the joint meeting of the Weighing and Software Sectors and possible future inclusion into Appendix D of NIST Handbook 44:

software-based devices: devices used to compute and control processes using software, where software is a general term for the programs and data used to operate the computers and/or related electronic devices. Software-based device may also consist of just software (e.g., weigh in/weigh out software).

Discussion/Conclusion

During the joint meeting of the Weighing and Software Sectors, the Chairman of the SS led a discussion on the identification of software; more specifically, the changes that have been proposed or that are needed to G-S.1. and G-S.1.1. and the reasons why these changes are important. He reviewed the SS's 2013 draft proposal to amend G-S.1. and G-S.1.1. and the comments that had been received since its distribution. Very few constructive comments had been received except for some comments provided by NIST OWM, which the Chairman reviewed one by one; requesting additional clarification from the NIST Technical advisor as needed.

Once the review of the Sector's draft proposal had been completed, it was then pointed out that NIST OWM's LMDG had developed some suggested alternative changes to the SS's proposal at the request of the SS. Members of both sectors were asked to review and consider the alternative changes proposed by OWM's LMDP, which were provided in a handout to members of both sectors and displayed on screen.

The NIST Technical Advisor to the WS, also a member of OWM's LMDP, explained the reasons for OWM's proposed alternative changes to *G-S.1. - Identification.* Initial discussions of the group regarding OWM's draft changes mostly concentrated on three main issues/concerns as follows:

1. Why is it necessary to retain the term "not-built-for-purpose software-based devices" and add enforcement dates to G-S.1.(d) when it is the Sector's intention to treat built-for-purpose and not-built-for-purpose devices the same with respect to identifying software?
2. Consideration of the text that OWM had developed and was proposing for addition to G-S.1.(d) iii.
3. What would be the effective dates of any changes agreed upon by the group?

The following is a brief summary of the discussions and actions taken by the two sectors relative to these three issues/concerns:

1. With regard to the changes proposed to G-S.1.(d), the NIST Technical Advisor to the WS indicated that it was OWM's view that a separation between built-for purpose and not-built-for-purpose software-based devices needed to be maintained within the paragraph because the current requirement (i.e., G-S.1.(d)) only applies to not-built-for-purpose software-based devices. Although the SS's intention is to expand the requirement to apply to all electronic devices, it would not be appropriate to require existing built-for-purpose-equipment, which is already in service, to comply with the proposed changes to G-S.1. since this equipment has not had to do so previously. Updating existing equipment, in order to make it comply with new requirements, could be costly to both manufacturers and device owners. Additionally, it may not be possible for some built-for-purpose devices to provide an indication of the current software version or revision identifier. Although marking of the version or revision identifier using a label affixed to the device might be an option, how would officials be able to tell if the version of software installed in the device actually matched the marking on the device? By adding effective dates, as proposed, the separation can be maintained and still provide a means of requiring all new electronic equipment to comply. The NIST Technical Advisor also acknowledged that it may be possible at some future date to remove the reference to "not built for-purpose" in the paragraph. Members of the two sectors agreed, although it was decided that the words "through December 31, 2015" in the lead-in sentence of G-S.1.(d) should be deleted because the inclusion of this date is not necessary and its removal does not in any way change the proposal.

2. There were significant concerns raised by equipment manufacturers regarding OWM's suggested proposal to require the continuous display of the version or revision identifier on software-based equipment having a digital display. It was stated that some displays; specifically referenced were "seven-segment digital displays of simple design," do not have the capability of complying with the proposed note that had been developed by OWM. It was also stated that customer demand for these simple displays remains steady among the different scale manufacturers because of their low cost in relation to other digital displays that incorporate more current and complex technology. That is, some customers aren't willing to pay the extra money for a more complex display that can be made to comply with OWM's proposed note, such as one of the graphic types, when all that's needed is a simple basic display. Manufacturers did not see this situation changing and stated that sales of these displays are driven by their low cost. Another concern was the valuable "real estate" that the version or revision identifier would take up if it were continuously displayed.

3. In consideration of the fact that the proposed changes, if adopted, would require both built-for purpose and not-built-for-purpose software-based equipment to continuously display the current software version or revision identifier or that this information be accessible via the display menus, members of the two sectors felt that the 2016 effective date proposed by OWM did not provide enough lead-in time for equipment manufacturers. Thus, the sectors agreed to extend the date to 2020 by amending OWM's proposal to reflect this new date.

A fourth issue/concern, which was raised by an equipment manufacturer somewhat later in the discussions, is that some built-for-purpose equipment have limited capability of displaying letters of the alphabet, and therefore, unable to comply with the prefacing requirements specified in G-S.1.(d)(1) and G-S.1.(d)(2). The example provided was a seven-segment display. It is not able to display a "V" or an "R," which are the current acceptable abbreviations for "version" and "revision," respectively. A "U" could be considered a symbol; however, it is not currently a symbol included in the list of acceptable abbreviations found in some *NCWM Publication 14* device checklists. Alternatively, a lower-case "r" could be displayed on such an indicator. In consideration of this concern, it was suggested that a "note" be added to G-S.1.(d) permitting the NTEP evaluators to specify a different method of indication if the device is incapable of prefacing the software version/revision with a "V" or "R." The sectors agreed to propose a "note" be added and let the S&T Committee decide whether the "note" is necessary or appropriate. An additional change agreed upon by the sectors relating to this issue/concern was to add the last sentence of G-S.1.(b) to the end of G-S.1.(d)(2). In discussing this issue/concern, it was also stated that some built-for-purpose devices only indicate the software version or revision identification during power up. That is, in order to view the software identification, it is necessary to shut off and then return power to the device. It was noted that some officials have been instructed not to power down equipment they are inspecting for liability reasons. There were no solutions to this (power down/power up) concern offered by members of either sector.

Although the SS had earlier proposed changes to G-S.1.1., it was decided during the meeting that no changes to G-S.1.1. were necessary since the sectors had agreed to retain the term "not-built-for-purpose software-based devices" in G-S.1.(d). Thus, no changes are proposed to paragraph G-S.1.1. The following reflects all of the changes to paragraph G-S.1. that were agreed upon by the two sectors during the joint meeting:

Amend NIST Handbook 44*:* G-S.1. Identification as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial

capitals, all capitals, or all lowercase.

[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(c) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and* ***~~not-built-for-purpose software-based software devices~~*** ***software****;*
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(d) the current software version or revision identifier for not-built-for-purpose software-based devices**; manufactured as of January 1, 2004 and all software-based devices or equipment manufactured as of January 1, 2020;**
~~[Nonretroactive as of January 1, 2004]~~
(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be:

i. *prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;*
[Nonretroactive as of January 1, 2007]
(Added 2006)

Note: If the equipment is capable of displaying the version or revision identifier but is unable to meet the formatting requirement, through the NTEP type evaluation process, other options may be deemed acceptable and described in the CC.
(Added 20XX)

ii. directly linked to the software itself; and
[Nonretroactive as of January 1, 2020]
(Added 20XX)

iii. continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable providing the device does not have an integral interface to communicate the version or revision identifier.
[Nonretroactive as of January 1, 2020]
(Added 20XX)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). ***Prefix lettering may be initial capitals, all capitals, or all lowercase.***
[Nonretroactive as of January 1, 2007]
(Added 2006)

(e) an National Type Evaluation Program (NTEP) Certificate of Conformance (CC) *number or a*

corresponding CC Addendum Number for devices that have a CC.

(1) *The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)*
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device. (Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~**and,**~~ 2006 **and 201X**)

An additional issue that was discussed during the joint meeting is whether or not the updating of metrological software should be considered a sealable event or sealable parameter. It was agreed that an update to metrological software is a sealable event and needs to be protected using an approved means of security. The sectors then considered whether it would be appropriate to include the updating of metrological software in the list of sealable parameters in *NCWM Publication 14* or to provide for its security by proposing a new General Code requirement be added to NIST Handbook 44. The sectors decided that the updating of metrological software can affect multiple sealable parameters, and therefore, it is appropriate to address its security in the General Code of *NIST Handbook 44*. Consequently, the sectors decided to complete and submit an NCWM Form 15 proposing there be a new General Code requirement added to the handbook to address the security of software updates.

The two sectors agreed that much progress had been made during the joint meeting, but that paragraph G-S.1., as revised during the meeting, is not likely to be considered for vote by the NCWM. In consideration of the progress that was made, the sectors agreed to recommend that the "Developing" status of the item be changed to "Informational" and forward the revised draft of G-S.1 to the different regional associations for their consideration at their next meeting.

Based on the feedback received by the S&T committee regarding agenda item 1, we are of the opinion that it may no longer be possible to avoid providing a definition for' software-based electronic devices'. A discussion on possible definitions commenced. Members of the two sectors reviewed a draft definition of "software-based devices" that had been developed by a member of the Sector in consideration of a comment that had been received by the S&T Committee during one of the 2014 NCWM Conferences. The Sectors decided that a simpler definition may be more palatable, e.g.:

Software Based Device – Any device with metrologically significant software.

If they feel it is imperative to have a definition for this term (which many in the Sector feel is self-defining), the S&T committee can point us in the direction of one or the other of the proposed definitions.

2. Identification of Certified Software

Source:
NTEP Software Sector

Background / Discussion:
This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?" In previous

meetings it was shown that the international community has addressed this issue (both WELMEC and OIML).

From WELMEC 7.2:

> **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**.**

From OIML D-31:

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the weights and measures inspector information that something has changed?
Yes, the Category III Audit Trail or other means of sealing.

How can the weights and measures inspector identify an NTEP Certified version?
They can't, without adding additional requirements like what is described here, in conjunction with including the identifier on the CC).

The sector believes that we should work towards language that would include a requirement similar to the International Organization of Legal Metrology (OIML) requirement in *NIST Handbook 44*. It is also the opinion of the sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

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

From OIML:

> 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)

Initial draft proposed language: (G-S.1.1?)

NIST Handbook 44 (This has been written into G-S.1.d.3): Identification of Certified Software:

Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified by the version or revision number~~.~~ ~~The identification~~, and this identification ~~of the software~~ shall be ~~inextricably~~ directly and inseparably linked to the software itself. The version or revision number may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

From NCWM Publication 14:

Identification of Certified Software:

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~**The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.**~~

The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

From OIML D-31:

Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose.

The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.

The first sentence of the first paragraph above is already addressed in *NIST Handbook 44's* marking requirements.

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

*(d) the current software version or revision identifier for ~~**not built-for-purpose**~~ **software-based electronic** devices;*
[Nonretroactive as of January 1, 2004]
(Added 2003) **(Amended 20XX)**

(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.
[Nonretroactive as of January 1, 2007]
(Added 2006)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).

[Nonretroactive as of January 1, 2007]
(Added 2006)

(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Also the sector recommends the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc.). Could also consist of / contain checksum, etc. (crc32, for example)

There was some additional discussion on this item regarding where this new requirement was best located. It was suggested that the first sentence of G-S.1.d.(3) could be added as a clause to the base paragraph G-S.1(d) text, e.g. “*the current software version or revision identifier for* ~~***not built-for purpose***~~ ***software-based*** *devices, which shall be directly and inseparably linked to the software itself;*” .

It also was suggested that the second sentence in G-S.1.d. (3) might be more suitable for *NCWM Publication 14,* as it describes more ”how” than ”what” the requirement entails.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc.). It could also consist of / contain checksum, etc. (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions that are still outstanding:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to ”inseparably link” the identifier to the software?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

At the 2012 NTEP Software Sector Meeting, there was some discussion as to where the terminology regarding inextricably linking the software version or revision to the software itself belonged. At the moment, it is not incorporated in the proposed text for G-S.1. *NCWM Publication 14* may be a better option for the time being. This would be another item that would benefit from further explanation in a supplementary document.

One suggestion was this revision to G-S.1.d:

(d) ~~when metrologically significant software is employed,~~ the current software version or revision identifier, which shall be directly and inseparably linked to the software itself;~~, for not-built-for-purpose software-based electronic devices;~~

Alternatively, if the previously proposed new subsection G-S.1.d.3 from Item 1 is adopted, this concept could be inserted thus:

> *(3) The version or revision identifier shall be* directly and inseparably linked to the software itself and *accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:*

Several sector members were of the opinion that attempting to make this change at the same time as the earlier changes might be a difficult sell. Mr. Truex, NTEP Administrator, reiterated the necessity of baby steps.

The sector recommended adding the following to *NCWM Publication 14* and forward to NTEP Weighing, Measuring, and Grain Analyzer sectors for feedback:

Identification of Certified Software:

> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

Also, it was decided to forward the two alternate options for adding requirements for uniquely identifying software to the individual sectors:

One suggestion was this revision to G-S.1.d:

> (d) ~~when metrologically significant software is employed,~~ the current software version or revision identifier, which shall be directly and inseparably linked to the software itself;~~, for not-built-for-purpose software-based electronic devices;~~

Alternatively, if the previously proposed new subsection G-S.1.d.3 from Item 1 is adopted, this concept could be inserted thus:

> *(3) The version or revision identifier shall be* **directly and inseparably linked to the software itself and** *accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an exception, permanently marking the version or revision identifier shall be acceptable under the following conditions:*

The Measuring Sector reviewed this item and had no feedback other than a statement that they support the continuing / ongoing efforts of this sector. The Weighing Sector summary mentioned that no one opted to provide comment. They agreed to take no further action on this item, pending further action from the Software Sector. This was specifically in reference to the accepted symbols.

For the time being, Jim Truex recommended that we not attempt to provide a definition for "software-based device". We discussed the possibility of combining this change with the first agenda item, which had been attempted in previous years. Alternatively, if the HB44 changes from agenda item 1 are made, this agenda item could be addressed in Pub. 14.

After further discussion, the proposed wording in G-S.1.d under agenda item 1 was changed. Agenda item 2 will remain; however, it will address potential changes to Pub. 14 and contain no suggested modifications to Handbook 44. (See changes and conclusion under agenda item 1 for further details.)

The Sector chair volunteered to review the existing slide presentation detailing the purpose of these changes, to ensure that it accurately reflects this information. This was done by the Technical Advisor and the most recent version reflects our current point of consensus (see Appendix B).

The list of acceptable menu text and symbols in Appendix A are intended to assist the labs in finding the certification number. The sector noticed no action by the sectors had been taken when this list was circulated for comment. We would like to remind them that we would like to have it reviewed. We feel that this belongs in, for example, the Weighing Device Pub. 14, page DES-22, Section 3; the Belt – Conveyor Scales, page BCS-10, Section 8.7; the Measuring Devices, page LMD-21, Section 1.6; the Grain Moisture Meter, page GMM-14, Section 1 (G.S.1); and Near Infrared Grain Analyzers, page NIR-8, Section 1 (G.S.1).

Conclusion:
Some of the sectors (Weighing, Measuring) have already agreed to put the two paragraphs of text appearing at the top of page NTEP-3 in Pub. 14. (The sentence that has been struck out in the first paragraph was not included because Handbook 44 hasn't been altered to make it a requirement.)

This agenda item will likely require less time during future meetings as it seems to be nearly finalized. Outstanding work remaining is to secure buy-in from the remaining sectors that have yet to adopt this recommendation to include in Pub. 14. Once those Sectors reach a decision, this item can be considered complete and removed from future Agendas.

3. Software Protection / Security

Source:
NTEP Software Sector

Background:
The sector agreed that *NIST 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 of development techniques have been applied.
>
> This requirement includes consideration of:
>
> 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 *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTEP Software Sector Meeting. 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 laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

The Maryland laboratory had particular questions regarding 3.1 and 5.1. The information for 3.1 could be acquired from an operator’s manual, a training video, or in-person training. The items in 5.1 were confusing to the evaluators. The terminology is familiar to software developers, but not necessarily others. It was indicated that manufacturers were typically quick to return the filled out questionnaire, but he didn’t know how his laboratory was supposed to verify that it was true. Generally, the laboratories wouldn't be expected to verify things to that level. For example, if the manufacturer states that a checksum is used to ensure integrity, the laboratories wouldn't be expected to evaluate the algorithm used.

The intent was to see whether the manufacturer had at least considered these issues, not for evaluators to become software engineers. Perhaps a glossary or descriptive paragraphs might be added to assist the evaluators for if the manufacturer has questions for the evaluators.

OIML makes use of supplementary documents to explain the checklist they use. Below are links:
http://www.oiml.org/publications/D/D031-e08.pdf
http://www.welmec.org/latest/guides/72.html
http://www.welmec.org/fileadmin/user_files/publications/2-3.pdf

WELMEC document 2.3 is the original source for our checklist, but it's been significantly revised and simplified. Mr. Payne, Maryland Department of Agriculture, is going to review the other documents and come up with some suggestions for the checklist. Mr. Roach, California Division of Measurement Standards, is going to begin using the checklist. The international viewpoint is that any device running an operating system is considered to be Type U. Mr. Roach mentioned that they're having lots of problems with "skimmers" stealing PIN's. Is there some way they can detect this?

Mr. Lewis, Rice Lake Weighing Systems, Inc., mentioned that he liked Measurement Canada's website. When answering similar questions, different pages would appear, based on answers to those questions: http://www.ic.gc.ca/eic/site/mc-mc.nsf/eng/lm00573.html

At the 2011 NTEP Software Sector Meeting, the laboratories were polled to obtain any feedback on the use of the checklist. Maryland attempted to use this checklist a few times. They had some difficulty obtaining answers from the manufacturers because the individual(s) interacting with the Maryland evaluator didn't always have the required information on hand. More experience in using the checklist will help determine what needs to be revised.

It was suggested that the checklist could be sent to manufacturers for their feedback as well, with the stipulation that it a completely voluntary exercise and purely informational at this point. The laboratories will coordinate with willing manufacturers to obtain feedback.

At the 2013 meeting, it was reported by the labs that attempts to use the current checklist did not meet with many difficulties. The checklists were given to the manufacturers to fill out, and that seemed to work rather well. Minor modifications were made to clarify certain confusing areas or eliminate redundancy (Note the text above includes the updates made in 2013).

Discussion:

The labs using this checklist on a trial basis indicated that there was some confusion as to versions/wording. There may be more than one version in circulation. The version shown in this Summary shall be used henceforth.

During the discussion, Ed Payne (NTEP lab, MD) said that his impression is that this is at least making some of the manufacturers think about security, which they hadn't necessarily done in the past.

It was indicated that some more or better examples may be helpful to manufacturers, and that more guidance is needed. Clearer instructions could be part of the checklist, or it could be a separate document. The Sector would like additional feedback specifically regarding what portions of it are causing confusion.

Due to proprietary issues, the labs can't simply give us direct feedback from the companies they interact with. Darrell Flocken volunteered to obtain information from the labs, aggregate it, and remove any potential proprietary information issues.

The checklist as updated during the 2014 meeting:

1. **Devices with Software**

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **<u>The manufacturer should indicate whether it's solely software or includes hardware in the system. Can the software be changed after the system has been shipped without breaking a seal?</u> AND** □ Yes □ No □ N/A

1.2. Cannot be modified or uploaded by any means after securing/verification. **<u>With the seal intact, can you change the software?</u>** □ Yes □ No □ N/A

Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, including version/revision. **<u>It may also include things like name, part number, CRC, etc.</u>** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. **<u>This means that you can't easily change the software without changing the software identifier. For example, the version identifier can't be in a text file that's easily editable, or in a variable that the user can edit.</u>** □ Yes □ No □ N/A

2. **Programmable or Loadable Metrologically Significant Software**

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. *The list of docs referred to exists in agenda item 5.* □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, Cyclical Redundancy Check (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. **Software with no access to the operating system and/or programs possible for the user. <u>This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.</u>**

3.1. 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

3.2. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User. Complete this section only if you replied No to 1.1.

4.1. 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 Weights and Measures jurisdiction and type-specific parameters). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

4.2. 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). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

5. Software Interface(s)

5.1. Verify the manufacturer has documented:

5.1.1. **If software separation is employed, t**he program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.1.2. **For software that can access the operating system or if the program is accessible to the user, t**he protective software interface itself is part of the metrologically significant software. □ Yes □ No □ N/A

5.1.3. The functions of the metrologically significant software that can be accessed ~~**via the protective software interface**~~. □ Yes □ No □ N/A

5.1.4. The **metrologically significant** parameters that may be exchanged ~~**via the protective software interface**~~ are defined. □ Yes □ No □ N/A

5.1.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.1.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

Conclusion:
The Sector discussed examples, such as the upgrade of application programs and how these changes would affect audit trails and version numbers. It should be clear that if the upgraded software doesn’t affect anything metrologically significant, then it’s irrelevant for the purposes of this checklist. On the other hand, if it does affect metrologically significant functions or parameters, it should be tracked and/or identified somehow.

The revised checklist will be reviewed and further edited as required, and the updated version can be sent to the labs.

4. Software Maintenance and Reconfiguration

Source:
NTEP Software Sector

Background:
After the software is completed, what do the manufacturers use to secure their software? The following items were reviewed by the sector. *Note that agenda Item 3 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 **<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 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 **<u>or become inoperative</u>**.

   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 weights and measures requirements.

The sector **agreed** that the two definitions below for Verified update and Traced update were acceptable.

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: It's possible that the Philosophy of Sealing section of NCWM Publication 14 may already address the above IF the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit).

Use of a Category 3 audit trail is required for a Traced Update. A log entry representing a traced software update shall include the software identification of the newly installed version.

The sector recommended consolidating the definitions with the above statement thus:

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**~~ Category 3 audit trail. The audit trail entry shall include the software identification of the newly installed version.

The sector recommended that as a first step, the following be added to *NCWM Publication 14*:

> **The updating of metrologically significant software, including software that checks the authenticity and integrity of the updates, shall be considered a sealable event.**

Mr. Truex, NTEP Administrator, believes the above sentence is unnecessary since it's self-evident. It was agreed to ask the other sectors for feedback on the value of this addition.

Though the sector is currently recommending only the single sentence be incorporated into *NCWM Publication 14* for the time being, ultimately, the sector may wish to advance the remaining language of the original item submission.

At the 2013 meeting, the Sector had no information indicating that the other sectors had yet been approached for feedback on the value of the addition of the proposed sentence. This sector would still like the other sectors to evaluate this for inclusion in Pub. 14. We'd also like to include some description indicating that an existing audit trail should be protected during a software update, though that may already be a requirement. This does appear to be addressed in the Requirements for Metrological Audit Trails Appendices in Pub. 14.

Discussion:
In 2010 the Software Sector had considered the following:

G-S.9. Metrologically Significant Software Updates

> The updating of metrologically significant software shall be considered a sealable event.
> Metrologically significant software that does not conform to the approved type is not allowed for use.

Ambler Thompson suggested that the notes under G-S.8. could be amended to include software updates as a new example. Rick Harshman recommended having it as a stand-alone item, such as discussed in 2010.

This could possibly be tied back to G-S.2.

What is the sealable parameter? Is it the software version / revision? Currently all of the parameters are user-selectable, which would make this unique.

If the general code in Handbook 44 is amended to include this in some form, it applies to everything. The various sectors don't need to add to their specific sections of Handbook 44.

Darrell Flocken suggested that we try to come up with a declaration of intent and see how the sectors respond. Doug Bliss will add it to the existing presentation. Jim Truex thought it might be valuable to obtain the opinion of the S&T Committee. The Legal Metrology group should be asked, "Is a software change that updates metrologically significant software a sealable event?" Rick Harshman can obtain an answer from them.

Ambler Thompson raised a concern about the fact that at this point none of the suggested wording requires that the software identifier be unique, i.e. a change to the metrologically significant software should require a change to the software identifier. You could perhaps infer it from the requirement that it be inextricably linked to the software, but that isn't clear. Jim Truex thinks this will eventually need to be addressed, but not right now.

We reviewed the presentation that Doug Bliss had revised and tweaked it a bit. This sparked more discussion about the difficulty of convincing NIST. There seems to be a fundamental difference in how they understand changes of parameters and/or software. People don't seem to understand the difference between software and data. Adding a slide that explains the difference may help.

Last year's Weighing Sector feedback (Jim Truex will provide their wording) – they were opposed because:

1. It would change the methods of sealing (category 1, 2, and 3 audit trails) and require a change to Handbook 44.
2. It's not clear that the requirement for authenticity and integrity of the updates is limited to metrologically significant software.

The other sectors were concerned about this as well.

Legacy equipment that's still being manufactured might need to be changed to meet this obligation since their audit trails wouldn't necessarily indicate that the software has been updated.

Reference G-S.8., which is rather loose. Pub. 14 goes into much more detail about what is metrologically significant.

Darrell Flocken referred to Handbook 44, the Scales code – the event logger category 3 – the software is not a parameter. It's not so much that the software would be tracked, as the fact that it has not been in the list of sealable parameters is the concern. It sounds like this may be a procedural issue – sections of Handbook 44 may need to be altered before the sectors can add this suggestion to Pub. 14.

Conclusion:
After the discussion during the 2014 joint meeting, we revised the wording of the proposed G-S.9 to reflect some of the concerns heard from the other Sectors and interested parties:

G-S.9. Metrologically Significant Software Updates

A software update that changes the metrologically significant software shall be considered a sealable event.

The Sector still feels that explicitly requiring the metrologically significant software to be given at least the same level of protection as metrologically significant parameters is the best approach. We look forward to feedback from the S&T Committee and other Sectors on this proposed change. The Software Sector still would like to consider the issue of audit trail protection; there is some doubt as to whether the existing language is sufficient as it does not address the integrity of the audit trail during a software update, etc.

5. NTEP Application for Software and Software-based Devices

Source:
NTEP Software Sector

Background:
The purpose of initiating this item was to identify issues, requirements and processes for type approving Type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, “The approval applicant is responsible for the provision of all the required equipment and components.” This would likely also be the policy of NTEP.

Discussion:
Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4): This is the list of documents referred to in the checklist.

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- ~~**A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms).**~~
- ~~**A description of the user interface, communication interface, menus, and dialogs.**~~
- The software identification (version, revision, etc.) and how to view it.
- ~~**An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc., if not described in the operating manual.**~~
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- ~~**The operating manual.**~~

Darrell Flocken and Jim Truex quickly reviewed existing requirements for documentation to be submitted for obtaining certification in Pub. 14, Administrative Policy, and on the application form itself. Administrative policy 9.1.7 was where this was found:

- Engineering specification
- Operating descriptions that characterize the type

NTEP evaluators already have the authority to request whatever documentation they need. We can provide them with a list of documents that we think would assist the evaluator in his job and also give the manufacturer a good idea of what they should be capable of providing.

Darrell Flocken suggested that these additional items on our list could be added to administrative policy 9.1.7 in Pub. 14. Jim Truex suggested it could also be added to the application.

We struck the second bullet point because the labs probably won't care about this particular issue since they already have tests that they'll be running to address the accuracy of the measuring algorithms.

Russ Vires suggested removing some of the other bullet points, reducing the list to only new things to be added to the administrative policy. The list was originally designed to replace the current required documentation, so this would change its purpose. The original list was also never intended to be all-inclusive.

Conclusion:
If we combine the two lists, it might appear as something like this:

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- A description of the user interface, communication interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc., if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.
- Engineering specification.
- Operating descriptions that characterize the type.

A statement could be made along the lines of, "If not included in the operating manual, provide the following, as applicable."

After the last sentence in 9.1.7, this could be added:

As part of the type evaluation submission, the following information should be provided for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.) and how to view it.**
- **An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.**

These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the labs is gathered. The Sector recommends including the above bulleted list as an introduction to the checklist as part of our recommendation to include the checklist from agenda item 3 in Pub. 14. As a description of the accuracy of the measuring algorithms, simply declaring the type and class being aimed for may be sufficient. This list should reflect the needs of the labs for an evaluation. The bulleted list and the paragraph before it should be brought to the labs for an initial review and their input.

The Sector needs to discuss any input from the labs and finalize this list, prior to submitting the list to the other Sectors for incorporation into Pub. 14.

6. Training of Field Inspectors

Source:
NTEP Software Sector

Background:
During discussions at the 2009 NTEP Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources

System Verification Tests
NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2

Example: On a vehicle scale, have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.
7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.

8. Over capacity.
8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]
8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.
9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)
10. Over capacity.
10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

The PDC is focused on training sessions at the moment, so it's unsure how much time they'd have to review this currently.

Discussion:
California has some direction for inspectors regarding third party software. Mike Wedman is currently tasked with revising and expanding some of California's documentation on the subject, and we asked him to share it with us when it is complete.

Is it California's Handbook 112? Mike Wedman said they don't have such a handbook; it's in their device enforcement documentation.

NIST Handbook 112 doesn't have anything specific to software, and Jim Truex says that this handbook has actually been out of production for years. Its last edition was in 2002. There's an online copy of it that was searched to verify if there was anything software-specific in it, and nothing was found.

Jim Pettinato proposed that we put together a group to begin writing something ourselves, and Jim Truex stressed that it needed to be written at a level that the field inspectors would find useful.

Conclusion:
We'll wait until Mike Wedman has completed his work on the California EPO. Jim Pettinato, Teri Gulke, and Mike Wedman volunteered to work on this offline.

The Sector would like to continue exploring means by which it can be of assistance in training of field inspectors as software and electronic systems become more and more prevalent in their daily tasks.

It was also suggested we contact Ross Anderson, a paid consultant working with the PDC committee, to ask his opinion on how the Software Sector could best proceed to assist in the training of field inspectors. The Sector chair, Jim Pettinato, will act as primary point of contact for this communication.

NEW ITEMS

7. Next Meeting

Background/Discussion:
The sector is on a yearly schedule for NTEP Software Sector Meetings. Now that we've adopted a joint meeting system, the next Sector joint meeting would likely be the Measuring Sector next October?

Conclusion:
The Measuring Sector normally meets the afternoon of Friday and all day Saturday, leading into Southern's meeting starting on Sunday. The labs meeting is Friday morning. Jim Truex recommended against us beginning Thursday and continuing into Friday as we will probably need an entire day overlapping with their schedule. Overlapping with them for the entire day of Saturday might be our best option, and then have a day to ourselves on Sunday.

We can't determine our precise schedule right now. We will have a one day meeting in conjunction with the Measuring Sector (in addition to a one day meeting of just the Software Sector), and it will be in the fall of 2015. Jim Truex is going to try to determine what's possible.

8. 2014 NCWM Interim Meeting Report

There was one item on the NCWM S&T Committee Agenda for the 2013 NCWM Interim Meeting related to work done by the NTEP Software Sector. *2013 Publication 15* S&T Item 360-2 relates to the 2013 NTEP Software Sector Agenda Item 1: Marking Requirements.

From Jim Truex – the S&T Committee reported that it is considering withdrawing the item from their agenda if the Software Sector doesn't show some progress this year. By the end of 8/28/14, this didn't seem like a likely result as we'd made significant progress on the item.

9. 2013 International Report

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the Sector. Software Sector Co-Chair Mr. Jim Pettinato will summarize the discussions that took place at the European Cooperation in Legal Metrology (WELMEC) WG7 meeting in Dec. 2013.

Highlights of interest to the NTEP Software Sector:

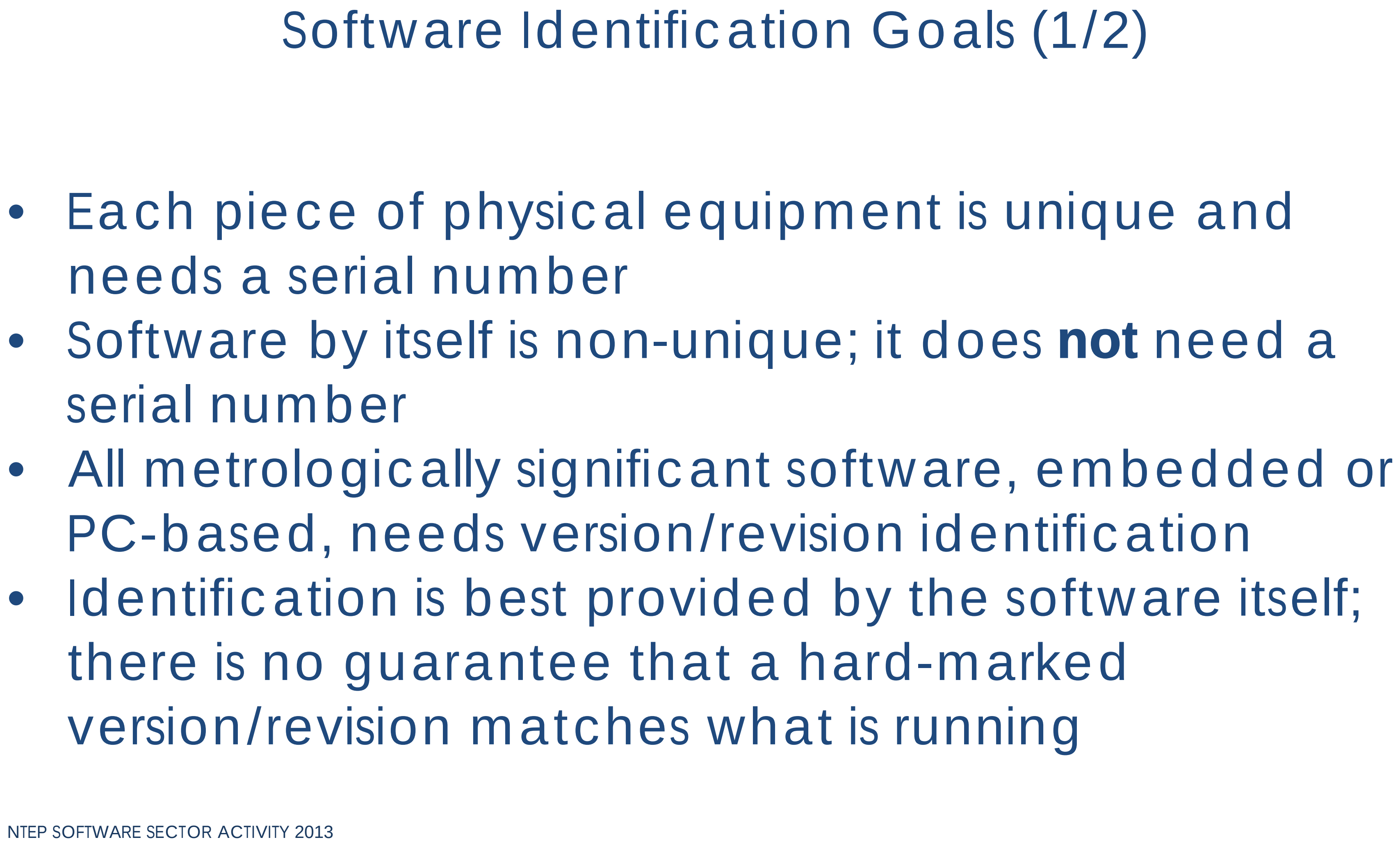
- New WELMEC 7.2 draft document circulated for comment by WG7
- R-117 working group

Appendix A – Acceptable Menu Text/Icons for Weights Measures information

Permitted Menu Text examples	*Permitted Icon shape examples*	*Essential characteristics*
Information Info	i i i	Top level menu text or icon • Icon text is a lower case "i" with block serifs • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Help ?	? ?	Top level menu text or icon • Icon text is a question mark • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Metrology Metrological Information	M M	Top or second level menu text or icon • Icon text is an upper case "M" • Text color may be light or dark but must contrast with the background color • Icon may have a circular, rectangular, or rounded rectangle border. • If present, the activation of this menu text/icon must recall at a minimum the NTEP CC number.
NTEP Data N.T.E.P. Certificate	NTEP NATIONAL CONFERENCE ON WEIGHTS AND MEASURES	This one is debatable – what if the certificate is revoked? Does NTEP grant holders of CCs the right to display the logo on the device, or just in documentation?
Weights & Measures Info	W&M W/M	

Appendix B: Software Sector 2014 Goals Presentation

SOFTWARE SECTOR 2014



Software Identification Goals (1/2)

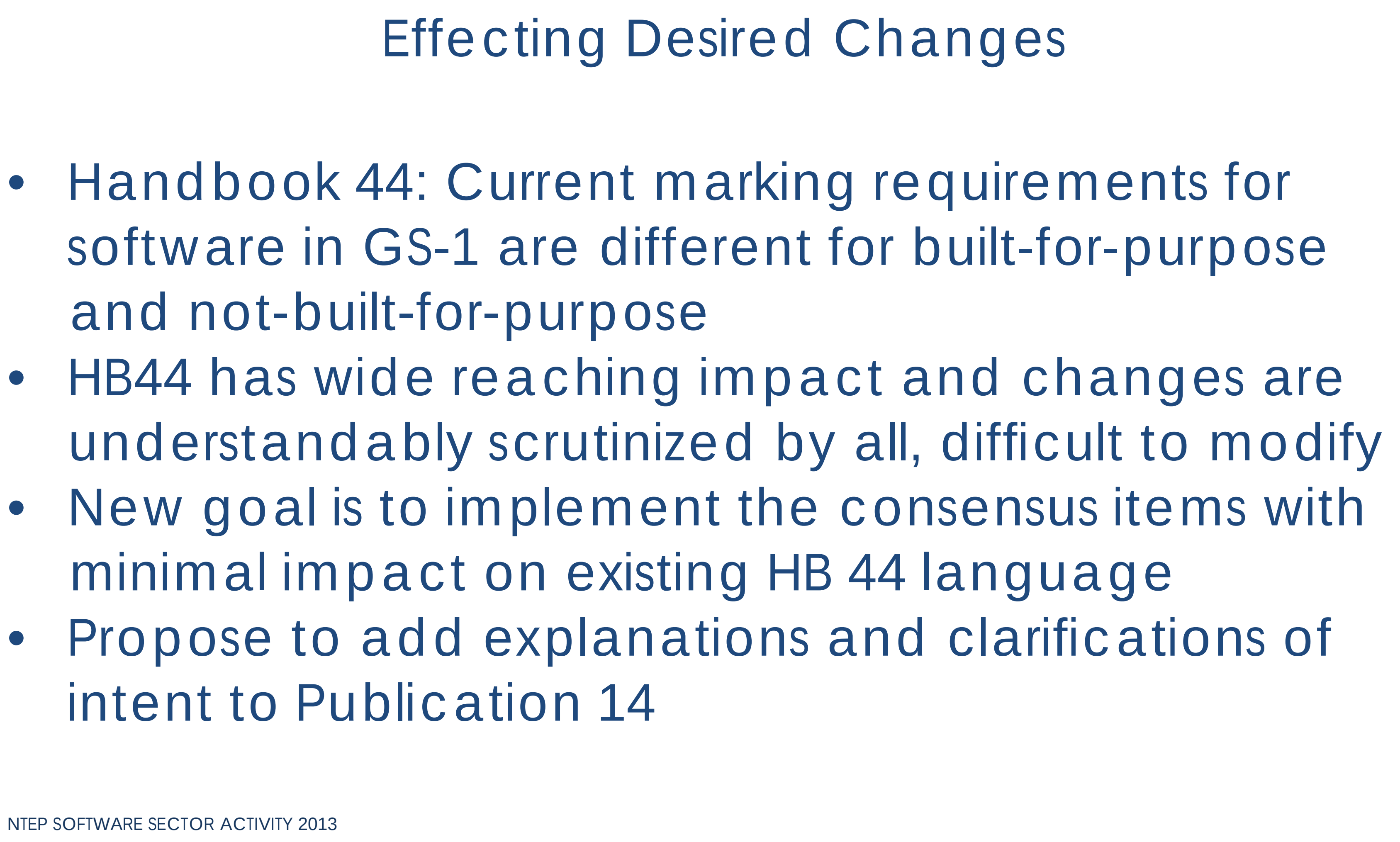
- Each piece of physical equipment is unique and needs a serial number
- Software by itself is non-unique; it does **not** need a serial number
- All metrologically significant software, embedded or PC-based, needs version/revision identification
- Identification is best provided by the software itself; there is no guarantee that a hard-marked version/revision matches what is running

NTEP SOFTWARE SECTOR ACTIVITY 2013

Software Identification Goals (2/2)

- Metrologically significant software and its version/revision identification must be linked together; it must not be possible to modify the software without a change to its identification and vice versa.
- Changes to metrologically significant software made after placement in service must be evident

NTEP SOFTWARE SECTOR ACTIVITY 2013



Effecting Desired Changes

- Handbook 44: Current marking requirements for software in GS-1 are different for built-for-purpose and not-built-for-purpose
- HB44 has wide reaching impact and changes are understandably scrutinized by all, difficult to modify
- New goal is to implement the consensus items with minimal impact on existing HB 44 language
- Propose to add explanations and clarifications of intent to Publication 14

NTEP SOFTWARE SECTOR ACTIVITY 2013

Software Identification

- Software must be identified, preferably self
- Handbook 44 proposed change:
  - Software identification must be displayable or printable, unless impossible (applies to all metrologically significant software)
- Publication 14 proposed additions:
  - Define software separation and explain options to submit software either as a monolithic entity that includes metrologically significant software or as a separated piece of metrologically significant software
  - Explain that metrologically significant software and its version/revision identifier must be linked together

NTEP SOFTWARE SECTOR ACTIVITY 2013

Recommended Additions to Publication 14

"Identification of Certified Software:

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.

The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrologically significant software and which does not."

NTEP SOFTWARE SECTOR ACTIVITY 2013

Software Protection

- Update of metrologically significant software must be protected
  - **Physical seal can protect software update but current event counters / audit trails may not**
  - **No clear requirement for counters/event log to either take note of, or survive a software update intact**
- Publication 14 proposed addition:
  - **Update of metrologically significant software becomes a sealable event**

Recommended Additions to Publication 14

“The updating of metrologically significant software, including software that checks the authenticity and integrity of the updates, shall be considered a sealable event.”

NTEP SOFTWARE SECTOR ACTIVITY 2013

Software Update

- Metrologically significant software contains algorithms, methods and procedures that operate on data, which includes both sealable and non-sealable parameters.
- Today, type approval evaluation considers protecting the modification of sealable parameters but ignores protecting the software that manipulates those sealable parameters.

Software Update (cont.)

- Equipment protected by a physical seal may prevent the update of software unless a seal is broken and provides evidence of software update.
- Event Counter & Event Logger sealing methods lack any requirement for such protection today.
- Software Sector believes that the field update of metrologically significant software is at least as important as the field change of a metrologically significant parameter – either can adversely impact a future measurement result.
- Metrologically significant software update should be a sealable event.

Future Vision

- Make Software Sector more visible /transparent
  - **Educate & better explain Software Sector objectives**
- Improve communication with other Sectors
  - **Propose to overlap Software Sector meetings with other Sector meetings to better align Publication 14 changes and speed up the consensus process**
- Finalize definition of ‘easily recognizable’ menu selections/icons to display software identification
- Provide checklists for software evaluations
- Assist in software-specific field training curriculum

NTEP SOFTWARE SECTOR ACTIVITY 2013

APPENDIX C – ATTENDEES

Doug Bliss
Mettler-Toledo, LLC
1150 Dearborn Drive
Worthington, OH 43085
P. (614) 438-4307 **F.** (614) 438-4355
E. doug.bliss@mt.com

Tom Buck
Ohio Department of Agriculture
8995 East Main Street
Reynoldsburg, OH 43068
P. (614) 728-6290 **F.** (614) 728-6424
E. tom.buck@agri.ohio.gov

Darrell Flocken
National Conference on Weights and Measures
1135 M Street, Suite 110
Lincoln, NE 68508
P. (614) 620-6134
E. darrell.flocken@ncwm.net

Andy Gell
FOSS North America
8091 Wallace Road
Eden Prarie, MN 55344
P. (952) 974-9892 **F.** (800) 547-6275
E. agell@fossna.com

Teri Gulke
Liquid Controls
105 Albrecht Drive
Lake Bluff, IL 60044-2242
P. (847) 283-8346 **F.** (847) 295-1170
E. tgulke@idexcorp.com

Tony Herrin
Cardinal Scale Manufacturing Co.
203 E. Daugherty
Webb City, MO 64870
P. (417) 673-4631
E. therrin@cardet.com

Paul A. Lewis, Sr.
Rice Lake Weighing Systems, Inc.
230 W. Coleman St.
Rice Lake, WI 54868
P. (715) 234-6967
E. plewis@ricelake.com

Edward McIntosh
F-RAMS, Inc.
P.O. Box 2964
Georgetown, TX 78627
P. (512) 868-8101
E. f-rams@mindspring.com

Eric Morabito
New York State W&M
10 B Airline Drive
Albany, NY 12206
P. (518) 457-3452
E. eric.morabito@agriculture.ny.gov

Christopher (Adam) Oldham
Gilbarco, Inc.
7300 West Friendly Avenue
High Point, NC 27420
P. (336) 547-5952
E. adam.oldham@gilbarco.com

Edward Payne
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
P. (410) 841-5790
E. edward.payne@maryland.gov

James M. Pettinato, Jr.
Senior Software Engineer
FMC Technologies, Inc.
1602 Wagner Ave.
P. (814) 898-5000
E. jim.pettinato@fmcti.com

Ambler Thompson
NIST, Office of Weights and MEsures
100 Bureau Drive, MS 20600
Gaithersburg, MD 21701
P. (301) 975-2333
E. ambler@nist.gov

Zacharias Tripoulas
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
P. (410) 841-5790 **F.** (410) 841-2765
E. zacharias.tripoulas@maryland.gov

Jim Truex
National Conference on Weights
and Measures
1135 M Street, Suite 110
Lincoln, NE 68508
P. (740) 919-4350 **F.** (740) 919-4348
E. jim.truex@ncwm.net

Mike Wedman
California Division of Measurement
Standards
6790 Florin Perkins Road, Suite 100
Sacramento, CA 95828
P. (916) 229-3014 **F.** (916)229-3026
E. mike.wedman@cdfa.ca.gov

Kraig Wooddell
Hobart Corporation
701 Ridge Avenue
Troy, OH 485374
P. (937) 332-2238
E. kraig.wooddell@hobartcorp.com

Note: The first day of the Software Sector meeting was held in conjunction with the NTEP Weighing Sector whose attendees were also present

National Type Evaluation Program (NTEP)
Software Sector Meeting Summary

September 16-17, 2015 / Denver, CO

INTRODUCTION

The charge of the National Type Evaluation Program (NTEP) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the National Type Evaluation Program (NTEP) Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be non-retroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OIML	International Organization of Legal Metrology
CC	Certificate of Conformance	OWM	Office of Weights and Measures
EPO	Examination Procedure Outline	PDC	Professional Development Committee
GMMs	Grain Moisture Meters	PDC	Professional Development Committee
NCWM	National Conference on Weights and Measures	S&T	Specifications and Tolerances Committee
NTEP	National Type Evaluation Program	SMA	Scale Manufactures Association
NTEP	National Type Evaluation Technical Committee	WELMEC	European Cooperation in Legal Metrology

Details of All Items
(In order by Reference Key)

WELCOME / INTRODUCTIONS

Since the first day of this year’s Sector meeting is a joint meeting with the Measuring Sector, there will be some time set aside to meet and greet both new and familiar faces. In addition, the Software Sector would like to give a brief presentation outlining the problems they've been asked to consider and some of the consensus that has been reached.

STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

Attendees of the 2015 NCWM Interim Meeting will be asked to share any relevant comments or discussion that took place during the open hearings or NCWM Standards and Tolerances (S&T) committee working sessions.

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), will provide a synopsis of international activity that relates to the work of the sector.

JOINT SESSION PROGRESS REPORT, ACTIVE ITEMS OF MUTUAL INTEREST

Since this is the first joint meeting of the Sectors, it is expected that some time will be required to review the agenda items of the Sectors that require collaboration, so all participants have a solid foundation for discussion. As part of this review, items of particular importance or interest should be allocated more time during the joint session day.

SOFTWARE SECTOR PRESENTATION

Doug Bliss, Sector Technical Advisor delivered our state-of-the-sector presentation to the joint meeting attendees.

CARRY-OVER ITEMS

1. Software Identification / Markings

Source:
NTEP Software Sector

Background / Discussion:
See the 2014 Software Sector Meeting Summary and the 2015 Interim Meeting S&T Agenda Item 310-1 for more background on this item.

Since its inception the sector has wrestled with the issue of software identification and marking requirements. At the 2014 meeting, significant work was done to make the recommendation to modify GS-1 more palatable to the Conference. The new approach was a less invasive modification with effective dates set in the future for compliance to new requirements.

Darrell Flocken reported on the discussions during the 2015 Interim meeting S&T Committee sessions. The item was left as a Developing item and was not officially commented upon during the session; the Committee indicated that they were waiting for the outcome from the joint meetings with the other sectors, especially this one, to move forward.

In 2015, in conjunction with the Measuring Sector, some additional fine tuning was done. The current recommendation is below.

Amend *NIST Handbook 44:* G-S.1. Identification as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(*c*) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and* ~~***not-built-for-purpose software-based software devices***~~ ***software****;*
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).

[Nonretroactive as of January 1, 2001]

(*d*) the current software version or revision identifier for not-built-for-purpose software-based devices~~;~~ **manufactured as of January 1, 2004 and all software-based devices or equipment manufactured as of January 1, 2022;**
~~***[Nonretroactive as of January 1, 2004]***~~
(Added 2003) **(Amended 2017)**

(1) The version or revision identifier shall be:

i. *prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;*
[Nonretroactive as of January 1, 2007]
(Added 2006)

Note: If the equipment is capable of displaying the version or revision identifier but is unable to meet the formatting requirement, through the NTEP type evaluation process, other options may be deemed acceptable and described in the CC.
(Added 2017)

ii. ***continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.***
[Nonretroactive as of January 1, 2022]
(Added 2017)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). ***Prefix lettering may be initial capitals, all capitals, or all lowercase.***
[Nonretroactive as of January 1, 2007]
(Added 2006) (Amended 2017)

(e) an National Type Evaluation Program (NTEP) Certificate of Conformance (CC) *number or a corresponding CC Addendum Number for devices that have a CC.*

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device. (Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~**and,**~~ 2006 **and 2017**)

Concerns were raised regarding situations where a particular device can be ordered with or without a display. In those situations, the manufacturers would prefer to hard-mark the software version/revision in all cases, keeping the manufacturing process simple. In this case, the wording "as an exception" is problematic since it is only allowed as an exception if the device has no capability of displaying it. Marc Buttler and Michael Keilty suggested that "exception" be replaced by "alternative", and "always" be added after "not" to address this concern, i.e.

iii. continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an ~~exception~~ alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.

[Nonretroactive as of January 1, 2022]
(Added 2017)

The Software Sector Chair asked the members of the Measuring and Software Sector in attendance whether everyone agreed to this modification of the proposal. Since no one objected, this change was included in the recommendation to the S&T Committee (and is included in the version shown above).

We debated whether to leave the non-retroactive date as 2020. It is possible to use 20XX and explain the intent in the proposal, but it might be better to leave it as a hard target. Since time has passed since we selected 2020, we backed it off until 2022, anticipating adoption by 2017 which would provide the intended period of five years after adoption.

In last year's proposal, there was an additional sub-clause included (in the 2014 Software Sector Summary version, this clause was in G-S.1.d(1).ii, and read ***directly linked to the software itself;)*** That line has been removed in this year's submission after further discussion during the 2015 joint meeting. Objections were raised that the clause did not actually represent a marking requirement. One suggestion was that it could be removed from Identification and moved to Sealing Requirements. Tina Butcher suggested instead that it be removed and a definition be added for Software Version or Revision Identifier. Unfortunately, if a definition is used instead the non-retroactive date would be lost. Another alternative suggested was to add a brand new section specifically for this; however, there's a general reluctance to add new sections to Handbook 44 that would have to be overcome.

It was realized that the word "permanently" in the very first paragraph of G-S.1 was sufficient language to require the software version or revision identifier to be linked to the software, so we ultimately decided to remove it from the proposed change.Since we already have a proposal on the agenda for the S&T Committee's meeting we will be submitting an amendment to reflect the new version of this proposal, rather than using Form 15 as for a new proposal.

This new version of the proposal has been sent to the various regions. Ideally, we should have someone at each of the regional meetings to answer any questions and champion this proposal.

Conclusion:
The amended proposal solves several areas of concern and has garnered consensus within multiple Sectors. We have forwarded the proposal to each of the Regional S&T committees and asked for consideration as a voting item; we also recommend that the Conference S&T committee consider making this a voting item in 2016.

2. Identification of Certified Software

Source:
NTEP Software Sector

Background:
This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?" In previous meetings it was shown that the international community has addressed this issue (both WELMEC and OIML).

From WELMEC 7.2:

> **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**.**

From OIML D-31:

> The executable file "**tt100_12.exe**" is protected against modification by a checksum. The value of checksum as determined by algorithm **XYZ** is **1A2B3C**.

Previous discussions have included a listing of some additional examples of possible valid methods (not limiting):

- CRC (cyclical redundancy check)
- Checksum
- Inextricably Linked version no.
- Encryption
- Digital Signature

Is there some method to give the weights and measures inspector information that something has changed?
Yes, the Category III Audit Trail or other means of sealing.

How can the weights and measures inspector identify an NTEP Certified version?
They can't, without adding additional requirements like what is described here, in conjunction with including the identifier on the CC).

The sector believes that we should work towards language that would include a requirement similar to the International Organization of Legal Metrology (OIML) requirement in *NIST Handbook 44*. It is also the opinion of the sector that a specific method should not be defined; rather the manufacturer should utilize a method and demonstrate the selected identification mechanism is suitable for the purpose. It is not clear from the discussion where such proposed language might belong.

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

From OIML:

> 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)

Identification of Certified Software:

> **Software-based electronic devices shall be designed such that the metrologically significant software is clearly identified by the version or revision number~~.~~ ~~The identification~~, and this identification ~~of the software~~ shall be ~~inextricably~~ directly and inseparably linked to the software itself. The version or revision number may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**

Discussion:

Some of the sectors have already agreed to put the below two paragraphs of text in the pertinent section(s) in Pub. 14. It is not yet reflected in the LMD and Vehicle Tank sections that are controlled by the Measuring Sector. The Measuring Sector was asked to consider inclusion of the paragraphs in 2014 as a sub-part of their Agenda Item 2, but it doesn't appear that it was specifically addressed.

From NCWM Publication 14:

Identification of Certified Software:

> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole. ~~**The conformity requirement applies to all parts and parts shall be marked according to Section G-S-X.X.**~~
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version or revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrologically significant software and which does not.

There was concern expressed related to the term "Certified Software" as it does not currently appear anywhere in Pub. 14 or Handbook 44. Jim Truex pointed out that this is intended as a note for Pub. 14, and "Certified" simply means that the software is traceable to a certificate.

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

> *(d) the current software version or revision identifier for ~~**not built-for-purpose**~~ **software-based electronic** devices;*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) **(Amended 20XX)**
>
> *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> *[Nonretroactive as of January 1, 2007]*
> *(Added 2006)*
>
> *(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).*
> *[Nonretroactive as of January 1, 2007]*
> (Added 2006)
>
> **(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**
> ***[Nonretroactive as of January 1, 201X]***
> **(Added 20XX)**

Also the sector recommended the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc. (crc32, for example)

Darrell Flocken shared his recollection of why the S&T Committee objected to this wording back in 2010. Basically, it went too deep for Handbook 44 and would be better placed in Pub. 14.

There was some additional discussion on this item regarding where this new requirement was best located. It was suggested that the first sentence of G-S.1.d.(3) could be added as a clause to the base paragraph G-S.1(d) text, e.g. "*the current software version or revision identifier for* ~~**not built for purpose**~~ ***software-based*** *devices, which shall be directly and inseparably linked to the software itself;"* .

It also was suggested that the second sentence in G-S.1.d. (3) might be more suitable for *NCWM Publication 14*, as it describes more "how" than "what" the requirement entails.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc. (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions previously brought up that have not really been satisfied to date are:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to "inseparably link" the identifier to the software?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

The possibility of creating a separate Publication 14 section specific to software was debated. There are pros and cons in terms of the chances of adoption with that approach. It might be beneficial to manufacturers, due to keeping the requirements in one place. This becomes a philosophical question – is the content of Handbook 44 intended to be a guide to manufacturers, or is it intended as direction to field inspectors? This discussion was tabled for present.

Historically, CC's have been written in terms of "version X and higher". It is not our intention to change that "policy", but it isn't documented anywhere. Perhaps that should be addressed by the Software Sector. Jim Truex reviewed the administrative policy text, which includes the requirement to report changes to NTEP, based on whether they're metrologically significant.

California indicated that their NTEP lab only puts the software version on the certificate if it's not-built-for-purpose, but it seems that the other labs do so for all software-based devices.

If pushed, the Sectors agreed that a simple defining statement to qualify the class of devices that are to be included would be forwarded to the interested parties:

> *Software Based Device – Any device with metrologically significant software.*

Conclusion:

The Software Sector decided that we'd leave the recommendation as-is, in the hopes that the changes to G-S.1 will be adopted at some point and then this can be revisited. Rich Miller, Marc Buttler, Dmitri Karimov, and the labs all indicated their support for the language as written.

The list of acceptable menu text and symbols in Appendix A are intended to assist the labs in finding the certification number. The sector noticed no action by the sectors had been taken when this list was circulated for comment. We would like to remind them that we would like to have it reviewed. We feel that this belongs in, for example, the Weighing Device Pub. 14, page DES-22, Section 3; the Belt – Conveyor Scales, page BCS-10, Section 8.7; the Measuring Devices, page LMD-21, Section 1.6; the Grain Moisture Meter, page GMM-14, Section 1 (G.S.1); and Near Infrared Grain Analyzers, page NIR-8, Section 1 (G.S.1).

- Tina Butcher mentioned that the Weighing Sector has a Weighing Checklist that has a similar set of approved symbols, so the examples shown in Appendix A would be in line with their current practice.

3. Software Protection / Security

Source:
NTEP Software Sector

Background / Discussion:
See the 2014 Software Sector Summary for additional background on this item.

The Sector continued to develop a proposed checklist for *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTEP Software Sector Meeting. 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 laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

The labs using this checklist on a trial basis indicated that there was some confusion as to versions/wording. There may be more than one version in circulation. The version shown in this Summary shall be used henceforth.

During the discussion, Ed Payne (NTEP lab, MD) said that his impression is that this is at least making some of the manufacturers think about security, which they hadn't necessarily done in the past.

It was indicated that some more or better examples may be helpful to manufacturers, and that more guidance is needed. Clearer instructions could be part of the checklist, or it could be a separate document. The Sector would like additional feedback specifically regarding what portions of it are causing confusion.

Due to proprietary issues, the labs can't simply give us direct feedback from the companies they interact with. Darrell Flocken volunteered to obtain information from the labs, aggregate it, and remove any potential proprietary information issues.

The checklist as updated during the 2014 meeting:

1. Devices with Software

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **The manufacturer should indicate** □ Yes □ No □ N/A

whether it's solely software or includes hardware in the system. Can the software be changed after the system has been shipped without breaking a seal? **AND**

1.2. Cannot be modified or uploaded by any means after securing/verification. **With the seal intact, can you change the software?** □ Yes □ No □ N/A

Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, including version/revision. **It may also include things like name, part number, CRC, etc.** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. **This means that you can't easily change the software without changing the software identifier. For example, the version identifier can't be in a text file that's easily editable, or in a variable that the user can edit.** □ Yes □ No □ N/A

2. Programmable or Loadable Metrologically Significant Software

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. *The list of docs referred to exists in agenda item 5.* □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, Cyclical Redundancy Check (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. Software with no access to the operating system and/or programs possible for the user. This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.

3.3. 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

3.4. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User. Complete this section only if you replied No to 1.1.

4.5. 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 Weights and Measures jurisdiction and □ Yes □ No □ N/A type-specific parameters). **<u>This is a declaration or explanation by the manufacturer.</u>**

4.6. 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). **<u>This is a declaration or explanation by the manufacturer.</u>** □ Yes □ No □ N/A

5. Software Interface(s)

5.7. Verify the manufacturer has documented:

5.7.1. **<u>If software separation is employed, t</u>**he program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.7.2. **<u>For software that can access the operating system or if the program is accessible to the user, t</u>**he protective software interface itself is part of the metrologically significant software. □ Yes □ No □ N/A

5.7.3. The functions of the metrologically significant software that can be accessed ~~**via the protective software interface**~~. □ Yes □ No □ N/A

5.7.4. The **<u>metrologically significant</u>** parameters that may be exchanged ~~**via the protective software interface**~~ are defined. □ Yes □ No □ N/A

5.7.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.7.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

The Sector discussed examples, such as the upgrade of application programs and how these changes would affect audit trails and version numbers. It should be clear that if the upgraded software doesn't affect anything metrologically significant, then it's irrelevant for the purposes of this checklist. On the other hand, if it does affect metrologically significant functions or parameters, it should be tracked and/or identified somehow.

Some of the labs have used the checklists, but they don't have easy access for the data to share. Not all of the labs have tried to use the checklist yet. In general, when the software programmers themselves are approached with the checklist, it's useful, but that's heavily dependent on who is interacting with the labs.

Jim Pettinato reiterated the Software Sector's request that the labs continue (or begin) to ask manufacturers whether they're willing to participate in the use of this checklist (on a voluntary basis), and to send their feedback to Darrell Flocken. Teri Gulke will clean up the checklist and put it in a separate document that can be posted on the NCWM website under the Software Sector's documents.

The contents of the checklist should tie back to requirements in Pub. 14. We originally crafted our checklist from the contents of D-31, so we went back to it to see if we could use it as a starting point for writing our own requirements for Pub. 14.

Though they need to be reworded, of course, the most useful portion of D-31 for our current purposes are probably sections 5.1.1., 5.1.3.2.a., 5.1.3.2.d, and 5.2.6.1. which state, respectively:

> *5.1.1 Software identification*
> *Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose. The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at*

> *start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.*
>
> *5.1.3.2.a The legally relevant software shall be secured against unauthorized modification, loading, or changes by swapping the memory device. In addition to mechanical sealing, technical means may be necessary to secure measuring instruments having an operating system or an option to load software.*
> *5.1.3.2.d Software protection comprises appropriate sealing by mechanical, electronic and/or cryptographic means, making an unauthorized intervention impossible or evident.*
>
> *5.2.6.1 Only versions of legally relevant software that conform to the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also depending on the kind of instrument under consideration.*

The question was asked, do these new requirements need to go into a new appendix specific to software in Pub. 14? Do we need to document new requirements at all if the checklist is put into Pub. 14? It could be considered that the checklist itself constitutes the new requirements. Darrell Flocken and Jim Truex supported that interpretation.

Conclusion:
The Sector asked that the revised checklist continue to be used by the labs. As we meet with each Sector jointly, we can get an updated report on the trial and decide if we're ready to recommend it for Pub. 14.

4. Software Maintenance and Reconfiguration

Source:
NTEP Software Sector

Background:
After the software is completed, what do the manufacturers use to install/secure/update their software? The following items were reviewed by the sector.

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

   A question from the floor, "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 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 weights and measures requirements.

The sector **agreed** that the two definitions below for Verified update and Traced update were acceptable.

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: It's possible that the Philosophy of Sealing section of NCWM Publication 14 may already address the above IF the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit).

Use of a Category 3 audit trail is required for a Traced Update. A log entry representing a traced software update shall include the software identification of the newly installed version.

The sector recommended consolidating the definitions with the above statement thus:

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**~~ Category 3 audit trail. The audit trail entry shall include the software identification of the newly installed version.

The sector recommended that as a first step, the following be added to *NCWM Publication 14*:

The updating of metrologically significant software, including software that checks the authenticity and integrity of the updates, shall be considered a sealable event.

Mr. Truex, NTEP Administrator, believes the above sentence is unnecessary since it's self-evident. It was agreed to ask the other sectors for feedback on the value of this addition.

Though the sector is currently recommending only the single sentence be incorporated into *NCWM Publication 14* for the time being, ultimately, the sector may wish to advance the remaining language of the original item submission.

At the 2013 meeting, the Sector had no information indicating that the other sectors had yet been approached for feedback on the value of the addition of the proposed sentence. This sector would still like the other sectors to evaluate this for inclusion in Pub. 14. We'd also like to include some description indicating that an existing audit trail should be protected during a software update, though that may already be a requirement. This does appear to be addressed in the Requirements for Metrological Audit Trails Appendices in Pub. 14.

Last year's Weighing Sector feedback indicated they were opposed because:

1. It would change the methods of sealing (category 1, 2, and 3 audit trails) and require a change to Handbook 44.
2. It's not clear that the requirement for authenticity and integrity of the updates is limited to metrologically significant software.

The other sectors were concerned about this as well.

Legacy equipment that's still being manufactured might need to be changed to meet this obligation since their audit trails wouldn't necessarily indicate that the software has been updated.

Reference G-S.8., which is rather loose. Pub. 14 goes into much more detail about what is metrologically significant.

Darrell Flocken referred to Handbook 44, the Scales code – the event logger category 3 – the software is not a parameter. It's not so much that the software would be tracked, as the fact that it has not been in the list of sealable parameters is the concern. It sounds like this may be a procedural issue – sections of Handbook 44 may need to be altered before the sectors can add this suggestion to Pub. 14.

In 2010 the Software Sector had considered the following:

<u>G-S.9. Metrologically Significant Software Updates</u>

> <u>The updating of metrologically significant software shall be considered a sealable event. Metrologically significant software that does not conform to the approved type is not allowed for use.</u>

Ambler Thompson suggested that the notes under G-S.8. could be amended to include software updates as a new example. Rick Harshman recommended having it as a stand-alone item, such as discussed in 2010.

This could possibly be tied back to G-S.2.

What is the sealable parameter? Is it the software version / revision? Currently all of the parameters are user-selectable, which would make this unique.

If the general code in Handbook 44 is amended to include this in some form, it applies to everything. The various sectors don't need to add to their specific sections of Handbook 44.

Darrell Flocken suggested that we try to come up with a declaration of intent and see how the sectors respond. Doug Bliss will add it to the existing presentation. Jim Truex thought it might be valuable to obtain the opinion of the S&T Committee. The Legal Metrology group should be asked, "Is a software change that updates metrologically significant software a sealable event?" Rick Harshman can obtain an answer from them.

Ambler Thompson raised a concern about the fact that at this point none of the suggested wording requires that the software identifier be unique, i.e. a change to the metrologically significant software should require a change to the software identifier. You could perhaps infer it from the requirement that it be inextricably linked to the software, but that isn't clear. Jim Truex thinks this will eventually need to be addressed, but not right now.

After the discussion during the 2014 joint meeting, we revised the wording of the proposed G-S.9 to reflect some of the concerns heard from the other Sectors and interested parties:

G-S.9. Metrologically Significant Software Updates

> **A software update that changes the metrologically significant software shall be considered a sealable event.**

The Sector still feels that explicitly requiring the metrologically significant software to be given at least the same level of protection as metrologically significant parameters is the best approach. We look forward to feedback from the S&T Committee and other Sectors on this proposed change. The Software Sector still would like to consider the issue of audit trail protection; there is some doubt as to whether the existing language is sufficient as it does not address the integrity of the audit trail during a software update, etc.

Discussion:

We debated once again whether this would be redundant. It can certainly be argued that G-S.8. already covers this requirement. If G-S.9. isn't added, is there support for changing Pub. 14 to add the software to the existing list of sealable parameters?

Philosophy of Sealing Appendix A in Pub. 14 doesn't specifically say anything about software. It discusses calibration and configuration parameters. There is a list of features and parameters that are typically sealed and another list of features and parameters that are not sealed. A note below states that these lists aren't fully inclusive, but anything that's metrologically significant does need to be sealed.

We've discussed before the fact that the terminology in Philosophy of Sealing repeatedly uses the term "parameter", which could cause confusion due to people interpreting this to only require sealing of parameters.

G-N.8. Checklist 2.18. for LND's in the Measuring Sector's Pub. 14 might be another place to add the word "software". This checklist is specific to the Measuring Sector's Pub. 14, so there wouldn't necessarily be something analogous in the other sectors' versions of Pub. 14.

G-S.8 refers to changing adjustable components, which could be interpreted as not having anything to do with software.

At one point the Software Sector had considered amending G-S.8., but that proved to be overly complicated.

Conclusion:

The Software and Measuring Sector attendees, as well as the lab representatives, have consensus that the proposed G-S.9 should be moved forward to the S&T Committee to be considered as a voting item in 2016. The Sector submitted this Recommendation for the addition of G-S.9 via Form 15.

5. NTEP Application for Software and Software-based Devices

Source:

NTEP Software Sector

Background:

The purpose of initiating this item was to identify issues, requirements and processes for type approving Type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, "The approval applicant is responsible for the provision of all the required equipment and components." This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4): This is the list of documents referred to in the checklist.

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.

Darrell Flocken and Jim Truex reviewed existing documentation required for obtaining certification in Pub. 14, administrative policy, and the application, to see what is already required. Administrative policy 9.1.7 was where this was found:

- Engineering specification
- Operating descriptions that characterize the type

NTEP evaluators already have the authority to request whatever documentation they need. We can provide them with a list of documents that we think would assist the evaluator in his job and also give the manufacturer a good idea of what they should be capable of providing.

Darrell Flocken suggested that this list could be added to administrative policy 9.1.7 in Pub. 14. Jim Truex suggested it could also be added to the application.

If we combine the two lists, it might appear as something like this:

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- A description of the user interface, communication interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.
- Engineering specification.
- Operating descriptions that characterize the type.

A statement could be made along the lines of, "If not included in the operating manual, provide the following, as applicable."

After the last sentence in 9.1.7, this could be added:
As part of the type evaluation submission, the following information should be provided for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.) , how to view it, and how it is tied to the software.**
- **An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.**

These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the labs is gathered. The Sector recommends including the above bulleted list as an introduction to the checklist as part of our recommendation to include the checklist from agenda item 3 in Pub. 14. As a description of the accuracy of the measuring algorithms, simply declaring the type and class being aimed for may be sufficient. This list should reflect the needs of the labs for an evaluation. The bulleted list and the paragraph before it should be brought to the labs for an initial review and their input.

There may be concerns with disclosure of proprietary information. Jim Truex says that the labs already protect other proprietary information. If the information provided is sufficiently high level, even theft of the data shouldn't cause too much of a concern.

Discussion:

The Measuring Sector Chair indicated that it was his opinion that it is not appropriate for the Measuring Sector, as a body, to make a recommendation regarding this proposal since it has to do with administrative policy.

According to Jim Truex, the labs already have the authorization to require this information.

While working on writing requirements for Pub. 14 from the checklist we've designed, we considered altering the second bullet point in our proposal for 9.17, so that it will require a description of how the software version or revision identifier is tied to the software itself.

Conclusion:
The Sector needs to discuss any input from the labs and finalize this list, prior to submitting the list to the other Sectors for incorporation into Pub. 14. The goal of this agenda item has somewhat shifted back to the original purpose, which is how do we communicate to applicants the expectations related to software based devices?

6. Training of Field Inspectors

Source:
NTEP Software Sector

Background:
During discussions at the 2009 NTEP Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources

System Verification Tests

NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2
   Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.
   7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.
8. Over capacity.
   8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]
   8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.
   9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)
10. Over capacity.
    10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Co-Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices.*

The PDC is focused on training sessions at the moment, so it's unsure how much time they'd have to review this currently.

Discussion:
It was suggested by Jim Truex and Darrell Flocken we make it part of our report as an attachment or an appendix of the meeting minutes. Then we can send out an email notifying the Software Sector members as to where to find it.

Alternatively, we could forward the document to the PDC Committee, tell them it was our starting point, and ask them for their suggestions.

The Sector would like to continue exploring means by which it can be of assistance in training of field inspectors as software and electronic systems become more and more prevalent in their daily tasks.

It was also suggested we contact Ross Anderson, a paid consultant working with the PDC committee, to ask his opinion on how the Software Sector could best proceed to assist in the training of field inspectors. The Sector chair, Jim Pettinato, will act as primary point of contact for this communication.

Conclusion:
The Sector would like to continue exploring means by which it can be of assistance in training of field inspectors as software and electronic systems become more and more prevalent in their daily tasks.

NEW ITEMS

7. Retrieval of Audit Log information

Source:
Adam Oldham, Gilbarco

Background:
The current requirements for a Category III audit trail include printing of log on demand. However, many devices are approved standalone and can be connected to systems that are approved standalone. How could Category 3 audit trail mechanisms be approved in situations where multiple devices need to work together to attain it? How can a device maintain Category 2 and 3 approvals in this scenario? What alternatives to printing can be considered as potentially valid solutions? (files, laptop, flash drive, etc).

Discussion:
This was discussed during the Measuring Sector's meeting on 9/15. The wording suggested was not agreed upon. Adam Oldham would like to have the Software Sector's suggestions, so he can put together a proposal for next year.

The US has rather unique requirements for printing the Category 3 audit trail, which are quite unwieldy – both in terms of the actual printing process (and results), as well as the needed approvals (the example provided by Adam

Oldham required an approval for each and every POS system that might be connected to their system). The most similar is from Mexico, but they require an electronic copy.

Darrell Flocken reported that there has been a little movement forward – alternative methods are now allowable, to some degree, but it's dependent on what the states are going to allow, and it still requires the ability to print it. The change will be in LMD Code S.2.2., not in Handbook 44 G-S.2.2.

We discussed the difficulty of requiring that the electronic data be printable on-site, given that some sites don't have any printers, and other sites may have printers attached to computers that are restricted in what can be used to attach to them.

In Mexico, Gilbarco relies upon laptops being present, supplied by the auditing company.

Darrell Flocken read the text of the actual changes that have been approved.

LMD Pub. 14 has a section in Appendix B Requirements for Metrological Audit Trails on the event logger, and that information doesn't seem to be in Handbook 44. In fact, it may even contradict what's in the LMD Pub. 14. In practice, what's in Pub. 14 tends to be more influential with evaluators.

Conclusion:

Adam Oldham will work on the wording for a proposal for next year that the Software Sector will review during the 2016 meeting.

8. Next Meeting

Background:

The sector is on a yearly schedule for NTEP Software Sector Meetings. Now that we've adopted a joint meeting system, the next Sector joint meeting will coincide with one of the remaining Sector meetings.

Discussion:

Belt Conveyor Scale Sector will meet in Feb. 2016 in Pittsburgh in conjunction with another association, and their meeting schedule does not allow for time on their agenda to discuss software issues. So this does not appear to be an option for a joint meeting.

Grain Analyzer didn't meet in 2015 because there were no new issues to discuss. We're not certain that they will have a meeting in 2016 either. They always meet in August in Kansas City at the same hotel, which doesn't have a meeting large enough to hold a joint meeting.

Conclusion:

Assuming the logistics can be worked out, the plan is to schedule a joint meeting with the Grain Analyzer Sector in Kansas City in 2016; ideally we'd like to have it in September.

9. 2015 NCWM Interim Meeting Report

There was one item on the NCWM S&T Committee Agenda for the 2015 NCWM Interim Meeting related to work done by the NTEP Software Sector:

- *2015 Publication 15* S&T Item 310-1 relates to the 2015 NTEP Software Sector Agenda Item 1: Marking Requirements.

The Committee concluded that the item would remain a Developing Item.

10. 2015 International Report

Dr. Ambler Thompson, NIST, Office of Weights and Measures (OWM), provided a synopsis of international activity that relates to the work of the Sector. Highlights of interest to the NTEP Software Sector:

OIML D31 is due to be updated, but there has been no activity as of yet. They have still not addressed field verification of software.

The terminology “inextricably linked” is under debate.

WELMEC 7.2 has superseded WELMEC 7.1.

Appendix A: Acceptable Menu Text/Icons for Weights & Measures information

Permitted Menu Text examples	*Permitted Icon shape examples*	*Essential characteristics*
Information Info	i i i	Top level menu text or icon • Icon text is a lower case "i" with block serifs • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Help ?	? ?	Top level menu text or icon • Icon text is a question mark • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Metrology Metrological Information	M M	Top or second level menu text or icon • Icon text is an upper case "M" • Text color may be light or dark but must contrast with the background color • Icon may have a circular, rectangular, or rounded rectangle border. • If present, the activation of this menu text/icon must recall at a minimum the NTEP CC number.
NTEP Data N.T.E.P. Certificate	NTEP NATIONAL CONFERENCE ON WEIGHTS AND MEASURES	This one is debatable – what if the certificate is revoked? Does NTEP grant holders of CCs the right to display the logo on the device, or just in documentation?
Weights & Measures Info	W&M W/M	

Appendix B: Final Attendee List

Doug Bliss
Mettler-Toledo, LLC
1150 Dearborn Drive
Worthington, OH 43085
P. (614) 438-4307 **F.** (614) 438-4355
E. doug.bliss@mt.com

Tom Buck
Ohio Department of Agriculture
8995 East Main Street
Reynoldsburg, OH 43068
P. (614) 728-6290 **F.** (614) 728-6424
E. tom.buck@agri.ohio.gov

Luciano Burtini
Measurement Canada
2008 Matera Avenue
Kelowna, BC V1V 1W9
P. (250) 862-6557
E. luciano.burtini@ic.gc.ca

Mario Dupuis
Measurement Canada
151 Tunney's Pasture Driveway
Ottawa, ON K1A 0C9
P. (613) 948-5009
E. mario.dupuis@ic.gc.ca

Joe Eccleston
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
P. (410) 841-5790
E. joseph.eccleston@maryland.gov

Darrell Flocken
National Conference on Weights and Measures
1135 M Street, Suite 110
Lincoln, NE 68508
P. (614) 620-6134
E. darrell.flocken@ncwm.net

Andrew Gell
FOSS North America
8091 Wallace Road
Eden Prairie, MN 55344
P. (952) 974-9892
E. agell@fossna.com

Dev Goyal
SICK USA
800 Technology Center Drive, Suite 6
Stoughton, MA 02072
P. (781) 302-2521
E. dev.goyal@sick.com

Teri Gulke
Liquid Controls, LLC
105 Albrecht Drive
Lake Bluff, IL 60044
P. (847) 283-8346
E. tgulke@idexcorp.com

Peter Kucmas
KROHNE
7 Dearborn Road
Peabody, MA 01960
P. (603) 497-7200
E. P.Kucmas@KROHNE.com

Edward McIntosh
F-RAMS, Inc.
3613 Williams Drive, Suite 603
Georgetown, TX 78628
P. (512) 868-8101 **F.** (512) 868-9115
E. f-rams@mindspring.com

Christopher (Adam) Oldham
Gilbarco, Inc.
7300 West Friendly Avenue
High Point, NC 27420
P. (336) 547-5952
E. adam.oldham@gilbarco.com

Edward Payne
Maryland Department of Agriculture
50 Harry S. Truman Parkway
Annapolis, MD 21401
P. (410) 841-5790
E. edward.payne@maryland.gov

James M. Pettinato, Jr.
FMC Technologies, Inc.
1602 Wagner Ave.
Erie, PA 16510
P. (814) 898-5000
E. jim.pettinato@fmcti.com

Steve Sharp
Liquid Controls, LLC
105 Albrecht Drive
Lake Bluff, IL 60044
P. (847) 283-8330
E. ssharp@idexcorp.com

Ambler Thompson
NIST, Office of Weights and Measures
100 Bureau Drive, MS 2600
Gaithersburg, MD 20899
P. (301) 975-2333
E. ambler@nist.gov

Jim Truex
National Conference on Weights and Measures
1135 M Street, Suite 110
Lincoln, NE 68508
P. (740) 919-4350 **F.** (740) 919-4348
E. jim.truex@ncwm.net

National Type Evaluation Program (NTEP) Software Sector Meeting Summary

September 14th, 2016 / Kansas City, MO

INTRODUCTION

The charge of the National Type Evaluation Program (NTEP) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the NTEP Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be non-retroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content | **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OIML	International Organization of Legal Metrology
CC	Certificate of Conformance	OWM	Office of Weights and Measures
EPO	Examination Procedure Outline	PDC	Professional Development Committee
NCWM	National Conference on Weights and Measures	S&T	Specifications and Tolerances Committee
NIST	National Institute of Standards and Technology	SMA	Scale Manufacturers Association
NTEP	National Type Evaluation Program	WELMEC	European Cooperation in Legal Metrology

Details of All Items
(In order by Reference Key)

I. SOFTWARE SECTOR PRESENTATION

Technical Advisor Doug Bliss gave a presentation from the Software Sector for the benefit of those Grain Analyzer Sector members who may not have been familiar with the agenda items and the background behind them. The presentation can be found on the NCWM.net web site for those interested in reviewing the background.

II. 2016 NCWM Interim and Annual Meeting Report

Darrell Flocken reported that the 2 Voting items from our Sector were passed by the Conference at the July meeting.

The marking requirement for Not Built for Purpose instruments begins January 1, 2017 and will begin to be required for Built for Purpose instruments in 2022. Diane Lee relayed Cathy Brenner's comment that she is only aware of one CC that has the software revision on it. One of the labs (GIPSA) checked the meters, and 2 out of 8 had the software revision on the label. Since built-for-purpose devices don't need to be able to indicate software revision until 2022, it is expected that the addition of this requirement will not pose a problem for grain analyzer manufacturers.

Also, in August Pub. 14 was revised by the Weighing Sector to include the requirement that changing software is a metrologically significant event.

III. 2016 International Activity Report

At the Berlin OIML TC5-SC2 meeting, Dr. Thompson met with Ulrich Grottker, who revealed a proposal to revise OIML D-31. He estimates it will take 3 – 5 years for the revision to be completed. Dr. Thompson suggested that the U.S. would volunteer to act as Secretariat for the document review process.

CARRY-OVER ITEMS

1. Software Identification / Markings

Source:
NTEP Software Sector

Background:
See the 2015 Software Sector Meeting Summary and the 2015 Interim Meeting S&T Agenda Item 310-1 for more background on this item.

Since its inception the sector has wrestled with the issue of software identification and marking requirements. At the 2014 meeting, significant work was done to make the recommendation to modify GS-1 more palatable to the Conference. The new approach was a less invasive modification with effective dates set in the future for compliance to new requirements.

Darrell Flocken reported on the discussions during the 2015 Interim meeting S&T Committee sessions. The item was left as a Developing item and was not officially commented upon during the session; the Committee indicated that they were waiting for the outcome from the joint meetings with the other sectors, especially this one, to move forward.

In 2015, in conjunction with the Measuring Sector, some additional fine tuning was done. The current recommendation is below.

Amend *NIST Handbook 44:* G-S.1. Identification as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(*c*) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and* ~~***not-built-for-purpose software-based software devices***~~ ***software****;*
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(*d*) the current software version or revision identifier for not-built-for-purpose software-based devices~~;~~

manufactured as of January 1, 2004 and all software-based devices or equipment manufactured as of January 1, 2022;
~~[Nonretroactive as of January 1, 2004]~~
(Added 2003) **(Amended 2017)**

(1) The version or revision identifier shall be:

i. *prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;*
[Nonretroactive as of January 1, 2007]
(Added 2006)

Note: If the equipment is capable of displaying the version or revision identifier but is unable to meet the formatting requirement, through the NTEP type evaluation process, other options may be deemed acceptable and described in the CC.
(Added 2017)

ii. ***continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.***
[Nonretroactive as of January 1, 2022]
(Added 2017)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). ***Prefix lettering may be initial capitals, all capitals, or all lowercase.***
[Nonretroactive as of January 1, 2007]
(Added 2006) (Amended 2017)

(e) an National Type Evaluation Program (NTEP) Certificate of Conformance (CC) *number or a corresponding CC Addendum Number for devices that have a CC.*

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device. (Amended 1985, 1991, 1999, 2000, 2001, 2003, **~~and,~~** 2006 **and 2017**)

Concerns were raised regarding situations where a particular device can be ordered with or without a display. In those situations, the manufacturers would prefer to hard-mark the software version/revision in all cases, keeping the manufacturing process simple. In this case, the wording "as an exception" is problematic since it is only allowed as an exception if the device has no capability of displaying it. Marc Buttler and Michael Keilty suggested that "exception" be replaced by "alternative", and "always" be added after "not" to address this concern, i.e.

iii. ***continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an ~~exception~~ alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.***

[Nonretroactive as of January 1, 2022]
(Added 2017)

The Software Sector Chair asked the members of the Measuring and Software Sector in attendance whether everyone agreed to this modification of the proposal. Since no one objected, this change was included in the recommendation to the S&T Committee (and is included in the version shown above).

We debated whether to leave the non-retroactive date as 2020. It is possible to use 20XX and explain the intent in the proposal, but it might be better to leave it as a hard target. Since time has passed since we selected 2020, we backed it off until 2022, anticipating adoption by 2017 which would provide the intended period of five years after adoption.

In last year's proposal, there was an additional sub-clause included (in the 2014 Software Sector Summary version, this clause was in G-S.1.d(1).ii, and read ***directly linked to the software itself;*** ***)*** That line has been removed in this year's submission after further discussion during the 2015 joint meeting. Objections were raised that the clause did not actually represent a marking requirement. One suggestion was that it could be removed from Identification and moved to Sealing Requirements. Tina Butcher suggested instead that it be removed and a definition be added for Software Version or Revision Identifier. Unfortunately, if a definition is used instead the non-retroactive date would be lost. Another alternative suggested was to add a brand new section specifically for this; however, there's a general reluctance to add new sections to Handbook 44 that would have to be overcome.

It was realized that the word "permanently" in the very first paragraph of G-S.1 was sufficient language to require the software version or revision identifier to be linked to the software, so we ultimately decided to remove it from the proposed change. Since we already have a proposal on the agenda for the S&T Committee's meeting we will be submitting an amendment to reflect the new version of this proposal, rather than using Form 15 as for a new proposal.

The new version of the proposal was sent to the regions and other Sectors for comment.

The amended proposal was Accepted as a Voting item at the 2016 Interim meeting and passed at the 2016 Annual Meeting.

Discussion:
Darrell Flocken reported that the Weighing Sector asked what alternatives were permissible (per the Note to G-S.1.d.i. above). Jim Pettinato described potential situations, such as a 7-segment display, where such a problem might exist.

Conclusion:
G-S.1.1. pertains to the location of marks. It currently maintains the distinction between built-for-purpose and not-built-for-purpose. The Software Sector would like to see that distinction eventually be eliminated, but the current thinking is that until the non-retroactive date of 2022 is reached, the differentiation cannot be eliminated. Due to that complication, Darrell Flocken's recommended we table the issue until then. Jim Truex pointed out that we should actually begin working on it in 2021 so it could be considered in 2022. The Software Sector agreed to remove this item from the agenda until that time. To prevent the intent to revisit this section of the general code being lost or forgotten, the item will remain on the agenda as a carry-over item that has been tabled until 2021.

2. Identification of Certified Software

Source:
NTEP Software Sector

Background:
See the 2015 Software Sector Meeting Summary for more background on this item.

This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?"

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

> *(d) the current software version or revision identifier for* ***~~not built for purpose~~*** ***software-based electronic*** *devices;*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) **(Amended 20XX)**
>
> *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> *[Nonretroactive as of January 1, 2007]*
> *(Added 2006)*
>
> *(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).*
> *[Nonretroactive as of January 1, 2007]*
> (Added 2006)
>
> **(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**
> ***[Nonretroactive as of January 1, 201X]***
> **(Added 20XX)**

Also the sector recommended the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc. (crc32, for example)

This item was eventually withdrawn. Darrell Flocken shared his recollection of why the S&T Committee objected to this wording back in 2010. Basically, it went too deep for Handbook 44 and would be better placed in Pub. 14.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc. (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions previously brought up that have not really been satisfied to date are:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to "inseparably link" the identifier to the software?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

The possibility of creating a separate Publication 14 section specific to software was debated. There are pros and cons in terms of the chances of adoption with that approach. It might be beneficial to manufacturers, due to keeping the requirements in one place. This becomes a philosophical question – is the content of Handbook 44 intended to be a guide to manufacturers, or is it intended as direction to field inspectors? This discussion was tabled for present.

Historically, CC's have been written in terms of "version X and higher". It is not our intention to change that "policy", but it isn't documented anywhere. Perhaps that should be addressed by the Software Sector. Jim Truex reviewed the administrative policy text, which includes the requirement to report changes to NTEP, based on whether they're metrologically significant.

California indicated that their NTEP lab only puts the software version on the certificate if it's not-built-for-purpose, but it seems that the other labs do so for all software-based devices.

If pushed, the Sectors agreed that a simple defining statement to qualify the class of devices that are to be included would be forwarded to the interested parties:

> *Software Based Device – Any device with metrologically significant software.*

The Software Sector decided that we'd leave the previously withdrawn recommendation as-is, in the hopes that the other changes to G-S.1 will be adopted and then this can be revisited. Several Measuring Sector members and all the labs indicated their support for the language as written.

Regarding field inspection and locating the required information: The list of acceptable menu text and symbols in Appendix A are intended to assist the labs in finding the certification number. The sector noticed no action by the sectors had been taken when this list was circulated for comment. We would like to remind them that we would like to have it reviewed. We feel that this belongs in, for example, the Weighing Device Pub. 14, page DES-22, Section 3; the Belt – Conveyor Scales, page BCS-10, Section 8.7; the Measuring Devices, page LMD-21, Section 1.6; the Grain Moisture Meter, page GMM-14, Section 1 (G.S.1); and Near Infrared Grain Analyzers, page NIR-8, Section 1 (G.S.1).

Tina Butcher mentioned that the Weighing Sector has a Weighing Checklist that has a similar set of approved symbols, so the examples shown in Appendix A would be in line with their current practice.

Discussion:
Since the G-S.1 change from Item 1 was voted on and adopted in 2016, we can now move forward on this item and consider adding to *NCWM Publication 14* the specifics that the Sector has been discussing related to presenting the software identification.

Darrell Flocken asked whether it's a specification or information. That would determine whether it should belong in HB44 or only in Pub. 14. One possibility is below:

(3) The version or revision identifier shall be directly and inseparably linked to the software itself.

Note: The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Concern was expressed that this could cause confusion with field inspectors. Software separation isn't something that's intended to be useful in the field, it is intended to ease type approval and software maintenance release processing. - This would lend weight to the argument of keeping it in Pub. 14.

If the Sector desires to include this in Pub. 14, we would need to identify all the sections where this concept would need to be added. The Software Sector doesn't have the authority to add it to the other sectors' Pub. 14's. Darrell Flocken reported that a note regarding the concept of software separation has already been added to several of the various Pub. 14 sections .

It was also noted that the checklist being developed for the labs currently includes (1.4.3) the requirement that the software version or revision be linked to the software itself.

Diane Lee relayed Cathy Brenner's comment that she believes that most grain analyzers are currently using a checksum, which would meet the requirement that the version/revision be linked to the software. The general consensus seemed to be that this type of requirement wouldn't be an imposition for grain analyzer manufacturers as it is already current practice to include a checksum.

As a side note, it was noted that there is precedence in the load cell code in HB44 of including requirements pertinent only at type evaluation. Darrell Flocken doesn't like this practice, but it is a possibility (for the requirement to make the software revision/version linked to the software itself).

Darrell Flocken found the wording added to Pub. 14 pertaining to the software version/revision marking requirement. The following wording has been added to the Weighing, Measuring, and Automatic Bulk Weighing sections of Marking Requirements (Section 3), but not the Grain Analyzer Sector's section because they hadn't had a meeting in 2015 (or the Near Infrared).

> 3. **Additional Marking Requirements- Not Built-for-Purpose Software-Based Devices**
> Identification of Certified Software:
>
> Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.
>
> The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

For the Weighing Sector, there is actually a holding spot in the checklist for this, due to the delay for implementation until 2022 for built-for-purpose. For now, it only pertains to not-built-for-purpose.

Darrell Flocken suggested that the text be rearranged a bit:

> 3. **Additional Marking Requirements- Not Built-for-Purpose Software-Based Devices**
> Identification of Certified Software:
>
> 3.1. The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is

comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not. Yes __ No ___ N/A ___

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.

Conclusion:

Jim Truex thinks that putting the requirement in the checklist in Pub. 14 could be linked to the marking requirement that was just adopted in 2016. Doug Bliss pointed out how permanence of markings are tested (via Pub. 14), but it isn't specifically spelled out in HB44.

Given that no grain analyzers are currently implemented as not-built-for-purpose devices, the requirement wouldn't affect them until 2022. Mr. Flocken will forward the proposed text to the other sectors (the Measuring Sector meets next week, but they have a full agenda already). Diane Lee will include this as part of the summary for Grain Analyzer's meeting, and ask for feedback and guidance as to where to put it. That means that it won't be adopted this year for the Grain Analyzer's section of Pub. 14.

The Chair proposed that we table Agenda Item 2 until 2021, and that we continue to pursue implementing the checklist in Pub. 14. Darrell Flocken suggested that the Software Sector make a recommendation that the various sectors adopt this for their Pub. 14's. It would take a year or so, to make it through all the various sectors. A note could be added saying that a device can't be rejected if it doesn't meet this requirement in the checklist until 2022. It was agreed that we would table this item until the 2021 meeting, at which time we will propose the following (updated) wording for the 2022 Pub. 14:

3. **Additional Marking Requirements- Software**

Identification of Certified Software:

The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.

3. Software Protection / Security

Source:
NTEP Software Sector

Background:
See the 2014 Software Sector Summary for additional background on this item.

The Sector continued to develop a proposed checklist for *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTEP Software Sector Meeting. 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 laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

The labs using this checklist on a trial basis indicated that there was some confusion as to versions/wording. There may be more than one version in circulation. The version shown in this Summary shall be used henceforth.

During the discussion, Ed Payne (NTEP lab, MD) said that his impression is that this is at least making some of the manufacturers think about security, which they hadn’t necessarily done in the past.

It was indicated that some more or better examples may be helpful to manufacturers, and that more guidance is needed. Clearer instructions could be part of the checklist, or it could be a separate document. The Sector would like additional feedback specifically regarding what portions of it are causing confusion.

Due to proprietary issues, the labs can’t simply give us direct feedback from the companies they interact with. Darrell Flocken volunteered to obtain information from the labs, aggregate it, and remove any potential proprietary information issues.

The checklist as updated during the 2014 meeting:

1. Devices with Software

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **The manufacturer should indicate whether it’s solely software or includes hardware in the system. Can the software be changed after the system has been shipped without breaking a seal? AND** □ Yes □ No □ N/A

1.2. Cannot be modified or uploaded by any means after securing/verification. **With the seal intact, can you change the software?** □ Yes □ No □ N/A

Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, including version/revision. **It may also include things like name, part number, CRC, etc.** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. **This means that you can't easily change the software without changing the software identifier. For example, the version identifier can't be in a text file that's easily editable, or in a variable that the user can edit.** □ Yes □ No □ N/A

2. Programmable or Loadable Metrologically Significant Software

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. *The list of docs referred to exists in agenda item 5.* □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, Cyclical Redundancy Check (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. Software with no access to the operating system and/or programs possible for the user. This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.

3.3. 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

3.4. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User. Complete this section only if you replied No to 1.1.

4.5. 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 Weights and Measures jurisdiction and type-specific parameters). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

4.6. 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). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

5. Software Interface(s)

5.7. Verify the manufacturer has documented:

5.7.1. **If software separation is employed, t**he program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.7.2. **For software that can access the operating system or if the program is accessible to the user, t**he protective software interface itself is part of the metrologically significant software. □ Yes □ No □ N/A

5.7.3. The functions of the metrologically significant software that can be accessed ~~via the protective software interface~~. □ Yes □ No □ N/A

5.7.4. The **metrologically significant** parameters that may be exchanged ~~**via the protective software interface**~~ are defined. □ Yes □ No □ N/A

5.7.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.7.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

The Sector discussed examples, such as the upgrade of application programs and how these changes would affect audit trails and version numbers. It should be clear that if the upgraded software doesn't affect anything metrologically significant, then it's irrelevant for the purposes of this checklist. On the other hand, if it does affect metrologically significant functions or parameters, it should be tracked and/or identified somehow.

Some of the labs have used the checklists, but they don't have easy access for the data to share. Not all of the labs have tried to use the checklist yet. In general, when the software programmers themselves are approached with the checklist, it's useful, but that's heavily dependent on who is interacting with the labs.

Jim Pettinato reiterated the Software Sector's request that the labs continue (or begin) to ask manufacturers whether they're willing to participate in the use of this checklist (on a voluntary basis), and to send their feedback to Darrell Flocken. Teri Gulke will clean up the checklist and put it in a separate document that can be posted on the NCWM website under the Software Sector's documents.

The contents of the checklist should tie back to requirements in Pub. 14. We originally crafted our checklist from the contents of D-31, so we went back to it to see if we could use it as a starting point for writing our own requirements for Pub. 14.

Though they need to be reworded, of course, the most useful portion of D-31 for our current purposes are probably sections 5.1.1., 5.1.3.2.a., 5.1.3.2.d, and 5.2.6.1. which state, respectively:

> *5.1.1 Software identification*
> *Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose. The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.*
>
> *5.1.3.2.a The legally relevant software shall be secured against unauthorized modification, loading, or changes by swapping the memory device. In addition to mechanical sealing, technical means may be necessary to secure measuring instruments having an operating system or an option to load software.*
> *5.1.3.2.d Software protection comprises appropriate sealing by mechanical, electronic and/or cryptographic means, making an unauthorized intervention impossible or evident.*
>
> *5.2.6.1 Only versions of legally relevant software that conform to the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also depending on the kind of instrument under consideration.*

The question was asked, do these new requirements need to go into a new appendix specific to software in Pub. 14? Do we need to document new requirements at all if the checklist is put into Pub. 14? It could be considered that the checklist itself constitutes the new requirements. Darrell Flocken and Jim Truex supported that interpretation.

The Sector asked that the revised checklist continue to be used by the labs.

As we meet with each Sector jointly, we can get an updated report on the trial and decide if we're ready to recommend it for Pub. 14. We can also look at the language from D-31 in more detail in an effort to craft guidance in line with NCWM/NTEP philosophy.

Discussion:
The Grain Analyzer Sector's labs have not had the opportunity to try using the checklist because they didn't meet in 2015. Tom Buck from Ohio reported that they've been giving the checklist to manufacturers but haven't been getting them back. Darrell Flocken has two examples, one for built-for-purpose and one for a built-for-purpose device.

Conclusion:
Jason Jordan from GIPSA said that they'd try it out. Doug Bliss and Jim Pettinato have volunteered to answer any questions that might arise as the labs attempt to use the checklist.

4. Software Maintenance and Reconfiguration

Source:
NTEP Software Sector

Background:
See the 2015 Software Sector Meeting Summary and the 2015 Interim Meeting S&T Agenda Item 310-1 for more background on this item.

After the software is completed, what do the manufacturers use to secure their software? The following items were reviewed by the sector. *Note that agenda Item 3 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 **<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 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 **<u>or become inoperative</u>**.

   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 weights and measures requirements.

The sector **agreed** that the two definitions below for Verified update and Traced update were acceptable.

> **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: It's possible that the Philosophy of Sealing section of NCWM Publication 14 may already address the above IF the definitions of Verified and Traced Updates (and the statement below) were to be added. The contrary argument was that it may be better to be explicit).

In 2010 the Software Sector had considered the following:

<u>G-S.9. Metrologically Significant Software Updates</u>
<u>The updating of metrologically significant software shall be considered a sealable event.</u>
<u>Metrologically significant software that does not conform to the approved type is not allowed for use.</u>

Ambler Thompson raised a concern about the fact that at this point none of the suggested wording requires that the software identifier be unique, i.e. a change to the metrologically significant software should require a change to the software identifier. You could perhaps infer it from the requirement that it be inextricably linked to the software, but that isn't clear. Jim Truex thinks this will eventually need to be addressed, but not right now.

After the discussion during the 2014 joint meeting, we revised the wording of the proposed G-S.9 to reflect some of the concerns heard from the other Sectors and interested parties:

<u>G-S.9. Metrologically Significant Software Updates</u>
<u>A software update that changes the metrologically significant software shall be considered a sealable event.</u>

The Sector still feels that explicitly requiring the metrologically significant software to be given at least the same level of protection as metrologically significant parameters is the best approach. We look forward to feedback from the S&T Committee and other Sectors on this proposed change. The Software Sector still would like to consider the issue of audit trail protection; there is some doubt as to whether the existing language is sufficient as it does not address the integrity of the audit trail during a software update, etc.

We debated once again whether this would be redundant. It can certainly be argued that G-S.8. already covers this requirement. If G-S.9. isn't added, is there support for changing Pub. 14 to add the software to the existing list of sealable parameters?

Philosophy of Sealing Appendix A in Pub. 14 doesn't specifically say anything about software. It discusses calibration and configuration parameters. There is a list of features and parameters that are typically sealed and another list of features and parameters that are not sealed. A note below states that these lists aren't fully inclusive, but anything that's metrologically significant does need to be sealed. We've discussed before the fact that the terminology in Philosophy of Sealing repeatedly uses the term "parameter", which could cause confusion due to people interpreting this to only require sealing of parameters. G-N.8. Checklist 2.18. for LND's in the Measuring Sector's Pub. 14 might be another place to add the word "software". This checklist is specific to the Measuring Sector's Pub. 14, so there wouldn't necessarily be something analogous in the other sectors' versions of Pub. 14. G-S.8 refers to changing adjustable components, which could be interpreted as not having anything to do with software. At one point the Software Sector had considered amending G-S.8., but that proved to be overly complicated.

The Software and Measuring Sector attendees, as well as the lab representatives agreed to forward the above proposed addition of G-S.9 to the S&T Committee and recommend it be considered as a voting item in 2016. This item (See 2016 Pub. 15, S&T Agenda Item 310-2) was voted upon and adopted at the 2016 Annual Meeting.

Discussion:
The Sector will decide if any further action on this item is required.

All currently approved grain analyzers provide Category 3 audit trails, and the Grain Analyzer Sector is planning to change HB44 to make it a requirement that all grain analyzers must be Category 3.

The Weighing Sector (which is the only Sector that’s met since the adoption of G-S.9.) has added language to Pub. 14’s Provision for Sealing, making software changes a sealable event.

Conclusion:
At this point, because the G-S.9 proposal has been voted upon and passed, the Software Sector can remove this item from its agenda. The only thing left to do is for the various Sectors to meet and adopt language similar to the Weighing Sector for their respective sections in Pub. 14.

5. NTEP Application for Software and Software-based Devices

Source:
NTEP Software Sector

Background:
The purpose of initiating this item was to identify issues, requirements and processes for type approving Type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, “The approval applicant is responsible for the provision of all the required equipment and components.” This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4): This is the list of documents referred to in the checklist.

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.

Darrell Flocken and Jim Truex reviewed existing documentation required for obtaining certification in Pub. 14, administrative policy, and the application, to see what is already required. Administrative policy 9.1.7 was where this was found:

- Engineering specification
- Operating descriptions that characterize the type

NTEP evaluators already have the authority to request whatever documentation they need. We can provide them with a list of documents that we think would assist the evaluator in his job and also give the manufacturer a good idea of what they should be capable of providing.

Darrell Flocken suggested that this list could be added to administrative policy 9.1.7 in Pub. 14. Jim Truex suggested it could also be added to the application.

If we combine the two lists, it might appear as something like this:

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- A description of the user interface, communication interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.

- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.
- Engineering specification.
- Operating descriptions that characterize the type.

A statement could be made along the lines of, "If not included in the operating manual, provide the following, as applicable."

After the last sentence in 9.1.7, this could be added:
As part of the type evaluation submission, the following information should be provided for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.) , how to view it, and how it is tied to the software.**
- **An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.**

These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the labs is gathered. The Sector recommends including the above bulleted list as an introduction to the checklist as part of our recommendation to include the checklist from agenda item 3 in Pub. 14. As a description of the accuracy of the measuring algorithms, simply declaring the type and class being aimed for may be sufficient. This list should reflect the needs of the labs for an evaluation. The bulleted list and the paragraph before it should be brought to the labs for an initial review and their input.

There may be concerns with disclosure of proprietary information. Jim Truex says that the labs already protect other proprietary information. If the information provided is sufficiently high level, even theft of the data shouldn't cause too much of a concern.

Michael Keilty didn't think it appropriate for the Measuring Sector, as a body, to make a recommendation regarding this proposal since it has to do with administrative policy.

According to Jim Truex, the labs already have the authorization to require this information.

While working on writing requirements for Pub. 14 from the checklist we've designed, we considered altering the second bullet point in our proposal for 9.17, so that it will require a description of how the software version or revision identifier is tied to the software itself.

Discussion:
The goal of this agenda item has somewhat shifted back to the original purpose, which is how do we communicate to applicants the expectations related to software based devices?

Diane Lee suggested we review the OIML requirements for documentation. The comment was made from the floor that OIML may go further than we are currently prepared to recommend.

Jason Jordan expressed his opinion that moving forward with this item will be helpful for the labs.

Darrell Flocken and Jim Truex think this should be added to the Application section. If limited to that section, it shouldn't require approval from any of the other Sectors.

Doug Bliss suggested that it might be easier to provide examples that do not meet acceptable standards.

As we began discussing the training of field inspectors, Darrell Flocken asked that we also provide further training for lab inspectors. There's an annual lab meeting typically around April, in 2017 it will be in Annapolis, MD.

Conclusion:
The Software Sector's recommendation will be to add the requirements to the Application section.

The Software Sector agreed to provide support for any desired training of lab personnel at the April meeting.

6. Training of Field Inspectors

Source:
NTEP Software Sector

Background:
During discussions at the 2009 NTEP Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources

System Verification Tests
NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2
   Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.

7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.

8. Over capacity.

   8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]

   8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.

   9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)

10. Over capacity.

    10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition, Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Co-Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices.*

The PDC is focused on training sessions at the moment, so it's unsure how much time they'd have to review this currently.

It was suggested by Jim Truex and Darrell Flocken we make it part of our report as an attachment or an appendix of the meeting minutes. Then we can send out an email notifying the Software Sector members as to where to find it.

Alternatively, we could forward the document to the PDC Committee, tell them it was our starting point, and ask them for their suggestions.

The Sector would like to continue exploring means by which it can be of assistance in training of field inspectors as software and electronic systems become more and more prevalent in their daily tasks.

It was also suggested we contact Ross Anderson, a paid consultant working with the PDC committee, to ask his opinion on how the Software Sector could best proceed to assist in the training of field inspectors. The Sector chair, Jim Pettinato, will act as primary point of contact for this communication.

Discussion:

Jim Pettinato hasn't heard anything further from the PDC or Ross Anderson, as they continue to be quite busy.

For the Grain Analyzer Sector, Diane Lee thought it would take some time to put together some training material, as they do not currently have anything in place for software requirements.

Examples from completed checklists would be very helpful when putting together field inspector training. A lot of training videos have been recently generated. Doug Musick suggested that we recommend adding this to the agenda for the PDC Committee. Certification exams could be updated more easily, on a state-by-state level. It might be better to make software a separate exam.

Diane Lee suggested we look at developing a basic course for software, incorporating specific guidelines for specific device types.

Amanda Dubin was concerned about having the field inspectors know all the different existing software, which is a monumental task. Instead, the training should focus on how to find the pertinent CoC and look up information from it on the website. Ideally, down the road there could be some sort of database or software tool disseminated to field inspectors to assist in the look up of certificate numbers and the approved version number(s) for the software for a particular device, and even instructions on how to view/print the audit trail.

Jim Truex holds a meeting once a year for the lab evaluators. Darrell Flocken suggested that we also focus on training them on software. Diane Lee mentioned that NIST has been having manufacturers coming in to provide training on, for example, how to access the audit trail.

Conclusion:
As mentioned in the previous agenda item, the lab meeting is expected to occur in the April timeframe next year and the Software Sector is willing to assist in providing such training.

Ambler Thompson will be reviewing the training courses to identify areas that will need to be updated to cover the new requirements that have been approved.

Jim Pettinato will contact Ross Anderson regarding the PDC Committee, offering the Software Sector’s assistance in continuing to develop training pertaining to software.

7. Retrieval of Audit Log information

Source:
Adam Oldham, Gilbarco

Background/Discussion:
The current requirements for a Category III audit trail include printing of log on demand. However, many devices are approved standalone and can be connected to systems that are approved standalone. How could Category 3 audit trail mechanisms be approved in situations where multiple devices need to work together to attain it? How can a device maintain Category 2 and 3 approvals in this scenario? What alternatives to printing can be considered as potentially valid solutions? (files, laptop, flash drive, etc).

This was discussed during the Measuring Sector's meeting on 9/15. The wording suggested was not agreed upon. Adam Oldham would like to have the Software Sector's suggestions, so he can put together a proposal for next year.

The US has rather unique requirements for printing the Category 3 audit trail, which are quite unwieldy – both in terms of the actual printing process (and results), as well as the needed approvals (the example provided by Adam Oldham required an approval for each and every POS system that might be connected to their system). The most similar is from Mexico, but they require an electronic copy.

Darrell Flocken reported that there has been a little movement forward – alternative methods are now allowable, to some degree, but it's dependent on what the states are going to allow, and it still requires the ability to print it. The change will be in LMD Code S.2.2., not in Handbook 44 G-S.2.2.

We discussed the difficulty of requiring that the electronic data be printable on-site, given that some sites don't have any printers, and other sites may have printers attached to computers that are restricted in what can be used to attach to them.

In Mexico, Gilbarco relies upon laptops being present, supplied by the auditing company.

LMD Pub. 14 has a section in Appendix B Requirements for Metrological Audit Trails on the event logger, and that information doesn't seem to be in Handbook 44. In fact, it may even contradict what's in the LMD Pub. 14. In practice, what's in Pub. 14 tends to be more influential with evaluators.

Adam Oldham will work on the wording for a proposal for next year that the Software Sector will review during the 2016 meeting.

Discussion:
Adam Oldham wasn't in attendance. Jim Pettinato reported that North Carolina had recently run into an instance where the audit trail wasn't printable on-site.

The devices monitored by the Grain Analyzer Sector are all Cat. 3, and they are all capable of printing the audit trail.

Doug Musick pointed out that if you keep the metrological information in the Cat. 3 audit trail, and separate that from the non-metrological information, there's less of a problem with the requirement to print the audit trail; however, such separation is not a requirement. Jim Pettinato discussed various options for limiting what's printed, such as selecting a date range.

Jim Pettinato reported that the S&T Committee reviewed this issue recently. Gilbarco's original proposal was shot down, but a revised proposal was made. Darrell Flocken reported that in July a version with the caveat that the inspector has discretion was voted upon and accepted.

Doug Bliss suggested we table this agenda item since we do not have a concrete proposal.

Conclusion:
Without a proposal and without Gilbarco being present, the Sector can take no action at this time. The Chair will attempt to ascertain whether the intent to move this item forward still exists prior to drafting next year's agenda.

NEW ITEMS

8. Transmission of Measurement Data

Source: Software Sector

Background:
General discussion on various issues related to distributed systems seen in use today and how metrology might be affected or vulnerable to facilitation of fraud. Specifically, authenticating sources of transactional data; guaranteeing integrity and/or retaining privacy of data; local vs. remote application functionality, SaaS.

Discussion:
The discussion began with an example: the integration of ‘smart’ utility meters that send data directly to the utility company and is designed to (eventually) eliminate the need to do local meter reading. In this application there may be need to associate data securely with the particular meter in question – be able to protect private information while guaranteeing the authenticity and integrity of the data being reported upstream.

Ambler Thompson discussed his experience with “smart metering”. They need some sort of positive ID, to associate the measurement with a time stamp, etc.

Doug Bliss pointed out that the Europeans have requirements on this subject, but they’re pulling back on them since there’s little they can do in field verification, type evaluation, etc. to actually enforce them.

Jim Pettinato asked the lab inspectors whether they regularly deal with systems that have portions remote from the originator of the data. Jim Truex responded that they deal with that all the time. Doug Musick says that he’s concerned particularly about retail fuel dispensers.

In the Grain Analyzer Sector, their inspectors typically check for issues by tracking individual transactions all the way down the data chain.

In instances of fraud, particularly man-in-the-middle attacks, the generation of fraud tends to be by the simplest means possible. Fraudsters at the current time seem generally to be attacking hardware, not software or communications interfaces. Also, it sounds like the various means of fraud are on a very case-by-case basis that would be impossible to apply across the board without major inconvenience to manufacturers.

Conclusion:
It sounds like it may be premature for the Software Sector to attempt to generate any recommendations or requirements on this subject. Jim Pettinato suggested that maybe at some point in time we could consider issuing some sort of statement on the subject, but not now.

We will remove this from future agendas. Jim Truex recommended that we not put it back in unless we get more specific requests to deal with the issue from other Sectors.

9. Use of GPS Receivers and Mapping Software for Trade (e.g. fare determination)

Source: Software Sector

Background:
Other committees have initiated conversation on this topic primarily due to the surge in popularity of alternate taxi services Uber and Lyft. Does the Software Sector see a need for technical guidance in this conversation? If so, what would be the scope of such guidance?

Discussion:
There were a few presentations at the Interim Meeting on this subject. The 2016 Annual Meeting archive (Denver 2016) has a presentation from Lyft that was given at that meeting.

Ambler Thompson has discussed this subject with the Europeans. One issue is traceability of the time stamp(s). You can also calculate velocity based upon the phase shift of the GPS signal, though it requires a high-end, survey-grade GPS receiver ($50k each). Car companies can use these devices to obtain a great deal of data.

Uber and Lyft claim that they are not billing upon GPS data, but rather a pre-negotiated contract based upon distance, time, and type of vehicle. Doug Bliss has been told that the bill is based upon the starting GPS location from the driver’s phone, the ending GPS location from the same phone, and a calculation of the shortest distance from Google Maps. If the driver’s phone doesn’t have a great GPS receiver, or if the reception is bad so it’s relying upon cell towers, etc., that’s a problem. We’re also not sure just how accurate Google Map’s route calculation is. Also, Google Maps is a disinterested third party whose database is being used for a purpose they didn’t specifically authorize.

Doug Musick reported that the Uber contract is based upon a unit price, though they do provide an estimate to the customer.

Jim Truex pointed out that the Taxi Meter Code in HB44 is obviously addressed to the old-style taxi. What’s in HB44 isn’t really applicable to the new Uber and Lyft paradigm.

John Barton is leading a working group dealing with the Taxi Meter Code.

Andrei Brezoica from California, who is on the working group, reported that there is a draft for new code to address this. Options exist for open-ended contracts for the customer. Google Maps is helping with the apps, pertaining to absolute distances, that Uber and Lyft are using. Jim Pettinato asked that Andrei Brezoica send us a copy of the draft recommendation.

Diane Lee pointed out that there are several exemptions elsewhere in the code, which may be useful as examples when working on changes to the Taxi Meter Code.

Doug Musick suggested that there could be a requirement for the companies to post their unit price, per-mile and per-time. Apparently Uber does this, but Lyft does not.

Conclusion:
The Software Sector will offer assistance to the working group dealing with the Taxi Meter code. Ambler Thompson will talk to John Barton.

10. Next Meeting

Background:
The sector is on a yearly schedule for NTEP Software Sector Meetings. Now that we've adopted a joint meeting system, the next Sector joint meeting will coincide with one of the remaining Sector meetings.

Discussion:
The Belt Conveyor Sector would be the next in the sequence, but they may not be a viable option. They may be meeting in November.

Jim Pettinato suggested that we instead schedule the Software Sector Meeting to convene with the Weighing Sector again. This would typically be in Annapolis, MD. The dates are still up in the air, but it would be close to Labor Day. The Grain Analyzer meeting is August 16 – 17. The Western Meeting also occurs in this timeframe.

The MDMD work group meeting might be another option, but it's in April, and they're not actually a sector. They meet in Columbus, OH. This could help us get on the agenda for each of the other sectors with any recommendations we might have for Pub. 14.

Jim Pettinato recommended we leave the decision up to Jim Truex and Darrell Flocken depending on logistics and availability of open dates.

Conclusion:
After reviewing potential scheduling conflicts in the August/September timeframe the group is leaning toward favoring the April option in conjunction with the MDMD meeting. Darrell Flocken will contact Robert Kensington (Chair of the MDMD Work Group) to verify that the MDMD work group would be okay with combining the meetings.

Appendix A – Acceptable Menu Text/Icons for Weights Measures information

Permitted Menu Text examples	*Permitted Icon shape examples*	*Essential characteristics*
Information Info	i i i	Top level menu text or icon • Icon text is a lower case "i" with block serifs • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Help ?	? ?	Top level menu text or icon • Icon text is a question mark • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Metrology Metrological Information	M M	Top or second level menu text or icon • Icon text is an upper case "M" • Text color may be light or dark but must contrast with the background color • Icon may have a circular, rectangular, or rounded rectangle border. • If present, the activation of this menu text/icon must recall at a minimum the NTEP CC number.
NTEP Data N.T.E.P. Certificate	NTEP	This one is debatable – what if the certificate is revoked? Does NTEP grant holders of CCs the right to display the logo on the device, or just in documentation?
Weights & Measures Info	W&M W/M	

Tina Butcher mentioned that the Weighing Sector has a Weighing Checklist that has a similar set of approved symbols, so the examples shown below would be in line with their current practice.

National Type Evaluation Program (NTEP)
Software Sector Meeting Summary

May 3rd, 2017 / Columbus, OH
(in conjunction with the Multiple Dimension Measuring Device work group)

INTRODUCTION

The charge of the National Type Evaluation Program (NTEP) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the NTEP Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be non-retroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OIML	International Organization of Legal Metrology
CC	Certificate of Conformance	OWM	Office of Weights and Measures
EPO	Examination Procedure Outline	PDC	Professional Development Committee
NCWM	National Conference on Weights and Measures	S&T	Specifications and Tolerances Committee
NIST	National Institute of Standards and Technology	SMA	Scale Manufacturers Association
NTEP	National Type Evaluation Program	WELMEC	European Cooperation in Legal Metrology

Details of All Items
(In order by Reference Key)

STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

NCWM Activity

The Software Sector didn't have any agenda items for the 2017 NCWM Interim Meeting. The Grain Analyzer Sector had an item regarding removable devices which did pertain to software. There were some suggested wording changes, to ensure it only covered metrologically-significant software, not all software changes. That item is Developing.

International Activity

Two weeks ago, we received a markup of D-31, which is being revised. Our concern is to ensure that the requirements are workable in the field. Originally the meeting was scheduled for June, but they've moved it back to mid-September in Berlin. Dr. Thompson intends to ask them to add a specific section for field inspectors. R129 was reviewed by the MDMD Work Group.

SOFTWARE SECTOR PRESENTATION

Technical Advisor Doug Bliss gave a presentation from the Software Sector for the benefit of those MDMD Work Group members who may not have been familiar with the Software Sector agenda items and the background behind them. The presentation can be found on the NCWM.net web site for those interested in reviewing the background.

CARRY-OVER ITEMS

1. Software Identification / Markings

Source:
NTEP Software Sector

Background:
See the 2016 Software Sector Meeting Summary for more background on this item.

Since its inception, the sector has wrestled with the issue of software identification and marking requirements. Numerous changes to the HB44 language were attempted and though support for the concepts was expressed, resistance to specific language made the course difficult. Finally, in 2015 in a joint meeting with the Measuring Sector, some additional fine tuning on the recommended changes to G-S.1 was done and we felt we had addressed everyone's concerns and had language ready to be voted upon for adoption. The recommended language is below.

Amend *NIST Handbook 44:* G-S.1. Identification as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(*c*) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and ~~**not built-for-purpose software-based software devices**~~ **software**;*
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(*d*) the current software version or revision identifier for not-built-for-purpose software-based devices~~;~~ **manufactured as of January 1, 2004 and all software-based devices or equipment manufactured as of January 1, 2022;**
~~***[Nonretroactive as of January 1, 2004]***~~
(Added 2003) **(Amended 2017)**

(1) The version or revision identifier shall be:

i. *prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;*
[Nonretroactive as of January 1, 2007]
(Added 2006)

Note: If the equipment is capable of displaying the version or revision identifier but is unable to meet the formatting requirement, through the NTEP type evaluation process, other options may be deemed acceptable and described in the CC.
(Added 2017)

ii. ***continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.***
[Nonretroactive as of January 1, 2022]
(Added 2017)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.*).* **Prefix lettering may be initial capitals, all capitals, or all lowercase.**
[Nonretroactive as of January 1, 2007]
(Added 2006) (Amended 2017)

(e) an National Type Evaluation Program (NTEP) Certificate of Conformance (CC) *number or a corresponding CC Addendum Number for devices that have a CC.*

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device. (Amended 1985, 1991, 1999, 2000, 2001, 2003, ~~**and,**~~ 2006 **and 2017**)

The amended proposal was Accepted as a Voting item at the 2016 Interim meeting and passed at the 2016 Annual Meeting.

Since the future work on this item depends on the expiration of the window for compliance (2022), the Sector agreed to table this item until 2020/2021, when we can again begin to discuss further modifications with the eventual goal of eliminating G-S.1.1 and the differentiation between built-for-purpose and not-built-for-purpose.

Discussion:
In July of 2016 the MDMD Work Group addressed some of these issues pertaining to software running on small devices such as phones that have very small screens. They discussed prioritization of what needed to be displayed, such as CC so that the remainder of the information can be looked up.

Conclusion:
This agenda item remains tabled until 2020.

2. Identification of Certified Software

Source:
NTEP Software Sector

Background:
See the 2016 Software Sector Meeting Summary for more background on this item.

This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?"

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

> *(d) the current software version or revision identifier for* ~~**not built-for-purpose**~~ ***software-based electronic*** *devices;*
> *[Nonretroactive as of January 1, 2004]*
> (Added 2003) **(Amended 20XX)**
>
> *(1) The version or revision identifier shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision.*
> *[Nonretroactive as of January 1, 2007]*
> *(Added 2006)*
>
> *(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.).*
> *[Nonretroactive as of January 1, 2007]*
> (Added 2006)
>
> **(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.**
> ***[Nonretroactive as of January 1, 201X]***
> **(Added 20XX)**

Also, the sector recommended the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.
- At a minimum, a version/revision indication (1.02.09, rev 3.0a, etc.). Could also consist of / contain checksum, etc. (crc32, for example)

This item was eventually withdrawn. Darrell Flocken shared his recollection of why the S&T Committee objected to this wording back in 2010. Basically, it went too deep for Handbook 44 and would be better placed in Pub. 14.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc. (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions previously brought up that have not really been satisfied to date are:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to “inseparably link” the identifier to the software?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

The possibility of creating a separate Publication 14 section specific to software was debated. There are pros and cons in terms of the chances of adoption with that approach. It might be beneficial to manufacturers, due to keeping the requirements in one place. This becomes a philosophical question – is the content of Handbook 44 intended to be a guide to manufacturers, or is it intended as direction to field inspectors? This discussion was tabled for present.

Historically, CC’s have been written in terms of “version X and higher”. It is not our intention to change that “policy”, but it isn’t documented anywhere. Perhaps that should be addressed by the Software Sector. Jim Truex reviewed the administrative policy text, which includes the requirement to report changes to NTEP, based on whether they’re metrologically significant.

California indicated that their NTEP lab only puts the software version on the certificate if it’s not-built-for-purpose, but it seems that the other labs do so for all software-based devices.

If pushed, the Sectors agreed that a simple defining statement to qualify the class of devices that are to be included would be forwarded to the interested parties:

Software Based Device – Any device with metrologically significant software.

The Software Sector decided that we’d leave the previously withdrawn recommendation as-is, in the hopes that the other changes to G-S.1 will be adopted and then this can be revisited. Several Measuring Sector members and all the labs indicated their support for the language as written.

Regarding field inspection and locating the required information: The list of acceptable menu text and symbols in Appendix A are intended to assist the labs in finding the certification number. The sector noticed no action by the sectors had been taken when this list was circulated for comment. We would like to remind them that we would like to have it reviewed. We feel that this belongs in, for example, the Weighing Device Pub. 14, page DES-22, Section 3; the Belt – Conveyor Scales, page BCS-10, Section 8.7; the Measuring Devices, page LMD-21, Section 1.6; the Grain Moisture Meter, page GMM-14, Section 1 (G.S.1); and Near Infrared Grain Analyzers, page NIR-8, Section 1 (G.S.1).

Tina Butcher mentioned that the Weighing Sector has a Weighing Checklist that has a similar set of approved symbols, so the examples shown in Appendix A would be in line with their current practice.

Since the recommended new G-S.1 language was voted on and adopted in 2016, we can now move forward on this item and consider adding to *NCWM Publication 14* the specifics that we have been discussing related to presenting the software identification.

Darrell Flocken asked whether it’s a specification or information. That would determine whether it should belong in HB44 or only in Pub. 14. One possibility is below:

(3) The version or revision identifier shall be directly and inseparably linked to the software itself.

Note: The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Concern was expressed that this could cause confusion with field inspectors. Software separation isn’t something that’s intended to be useful in the field, it is intended to ease type approval and software maintenance release processing. - This would lend weight to the argument of keeping it in Pub. 14.

If the Sector desires to include this in Pub. 14, we would need to identify all the sections where this concept would need to be added. The Software Sector doesn't have the authority to add it to the other sectors' Pub. 14's. Darrell Flocken reported that a note regarding the concept of software separation has already been added to several of the various Pub. 14 sections.

It was also noted that the checklist being developed for the labs currently includes (1.4.3) the requirement that the software version or revision be linked to the software itself.

The Chair proposed that we table Agenda Item 2 until 2021, and that we continue to pursue implementing the checklist in Pub. 14. Darrell Flocken suggested that the Software Sector recommend that the various sectors adopt this for their Pub. 14's. It would take a year or so, to make it through all the various sectors. A note could be added saying that a device can't be rejected if it doesn't meet this requirement in the checklist until 2022. It was agreed that we would table this item until the 2021 meeting, at which time we will propose the following (updated) wording for the 2022 Pub. 14:

3. **Additional Marking Requirements- Software**

Identification of Certified Software:

The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.

Discussion:
At the 2017 joint meeting, the MDMD Work Group discussed adding the section regarding linking of identifier to the software to their section in Pub. 14. There were no objections, so Darrell Flocken said he'd add it for next year's publication. A note shall be added that this is voluntary until 2022.

Also, we further discussed the idea of software separation, especially in how it pertains to the difference between the terms "metrologically significant" and "legally relevant". Some legal requirements have nothing to do with metrology. There is a difference in how the US regards this (since each state can have different legal requirements) vs. the philosophy in Europe. There isn't a definition of "metrologically significant" in Handbook 44, but Publication 14 has a description of all the parameters that needs to be sealed, which includes both metrologically significant and legally relevant parameters.

A definition of "metrologically significant" could be helpful, but Darrell Flocken suggested that we make sure it doesn't contradict VCAP's administrative policies.

Handbook. 44 does contain a definition for "metrological integrity".

Type evaluation is the time at which decisions are made regarding which exact parameters are sealable. According to Jim Truex, the US has never been able to come to a consensus on this subject.

Conclusion:
Jim Pettinato suggested that we work offline to generate a description intended to provide guidance on what we mean by “metrologically significant”. Jim Pettinato, Doug Bliss, Dr. Ambler Thompson, and Kevin Detert volunteered to make up a subcommittee to address this subject.

We also considered the issue of having to adopt a general software requirement to multiple sections of Publication 14 to address essentially the same requirement for each category of device separately. The idea was floated by the Sector that perhaps a new section should be added to Publication 14 specific to software that applies to all metrologically significant software in all devices types that might contain such. Rather than formally suggesting this be done, we decided to informally run the idea past the Specifications and Tolerances committee. That way, if there was little interest or strong objection, we wouldn’t waste time generating a draft.

3. Software Protection / Security

Source:
NTEP Software Sector

Background:
See the 2014 Software Sector Summary for additional background on this item.

The Sector continued to develop a proposed checklist for *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTEP Software Sector Meeting. 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 labs using this checklist on a trial basis indicated that there was some confusion as to versions/wording. There may be more than one version in circulation. The version shown in this Summary shall be used henceforth.

During the discussion, Ed Payne (NTEP lab, MD) said that his impression is that this is at least making some of the manufacturers think about security, which they hadn't necessarily done in the past.

It was indicated that some more or better examples may be helpful to manufacturers, and that more guidance is needed. Clearer instructions could be part of the checklist, or it could be a separate document. The Sector would like additional feedback specifically regarding what portions of it are causing confusion.

Due to proprietary issues, the labs can't simply give us direct feedback from the companies they interact with. Darrell Flocken volunteered to obtain information from the labs, aggregate it, and remove any potential proprietary information issues.

The checklist as updated during the 2014 meeting:

1. Devices with Software

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **The manufacturer should indicate whether it's solely software or includes hardware in the system. Can the software be changed after the system has been shipped without breaking a seal?** **AND** □ Yes □ No □ N/A

1.2. Cannot be modified or uploaded by any means after securing/verification. **With the seal intact, can you change the software?** □ Yes □ No □ N/A

Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, including version/revision. **It may also include things like name, part number, CRC, etc.** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. **This means that you can't easily change the software without changing the software identifier. For example, the version identifier can't be in a text file that's easily editable, or in a variable that the user can edit.** □ Yes □ No □ N/A

2. Programmable or Loadable Metrologically Significant Software

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. *The list of docs referred to exists in agenda item 5.* □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, Cyclical Redundancy Check (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. Software with no access to the operating system and/or programs possible for the user. This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.

3.3. 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

3.4. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User. Complete this section only if you replied No to 1.1.

4.5. 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 Weights and Measures jurisdiction and type-specific parameters). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

4.6. 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). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

5. Software Interface(s)

5.7. Verify the manufacturer has documented:

5.7.1. **If software separation is employed, t**he program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.7.2. **For software that can access the operating system or if the program is accessible to the user, t**he protective software interface itself is part of the metrologically significant software. □ Yes □ No □ N/A

5.7.3. The functions of the metrologically significant software that can be accessed ~~**via the protective software interface**~~. □ Yes □ No □ N/A

5.7.4. The **metrologically significant** parameters that may be exchanged ~~**via the protective software interface**~~ are defined. □ Yes □ No □ N/A

5.7.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.7.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

Jim Pettinato reiterated the Software Sector's request that the labs continue (or begin) to ask manufacturers whether they're willing to participate in the use of this checklist (on a voluntary basis), and to send their feedback to Darrell Flocken. Teri Gulke will clean up the checklist and put it in a separate document that can be posted on the NCWM website under the Software Sector's documents.

The contents of the checklist should tie back to requirements in Pub. 14. We originally crafted our checklist from the contents of D-31, so we went back to it to see if we could use it as a starting point for writing our own requirements for Pub. 14.

Though they need to be reworded, of course, the most useful portion of D-31 for our current purposes are probably sections 5.1.1., 5.1.3.2.a., 5.1.3.2.d, and 5.2.6.1. which state, respectively:

> *5.1.1 Software identification*
> *Legally relevant software of a measuring instrument / electronic device / sub-assembly shall be clearly identified with the software version or another token. The identification may consist of more than one part but at least one part shall be dedicated to the legal purpose. The identification shall be inextricably linked to the software itself and shall be presented or printed on command or displayed during operation or at start up for a measuring instrument that can be turned off and on again. If a sub-assembly/an electronic device has neither display nor printer, the identification shall be sent via a communication interface in order to be displayed/printed on another sub-assembly/electronic device.*
>
> *5.1.3.2.a The legally relevant software shall be secured against unauthorized modification, loading, or changes by swapping the memory device. In addition to mechanical sealing, technical means may be necessary to secure measuring instruments having an operating system or an option to load software.*
> *5.1.3.2.d Software protection comprises appropriate sealing by mechanical, electronic and/or cryptographic means, making an unauthorized intervention impossible or evident.*
>
> *5.2.6.1 Only versions of legally relevant software that conform to the approved type are allowed for use (see 5.2.5). Applicability of the following requirements depends on the kind of instrument and is to be worked out in the relevant OIML Recommendation. It may differ also depending on the kind of instrument under consideration.*

The question was again asked, do these new requirements need to go into a new appendix specific to software in Pub. 14? Do we need to document new requirements at all if the checklist is put into Pub. 14? It could be considered that the checklist itself constitutes the new requirements. Darrell Flocken and Jim Truex supported that interpretation.

At the 2016 meeting, we learned that the Grain Analyzer Sector's labs have not had the opportunity to try using the checklist because they didn't meet in 2015. Tom Buck from Ohio reported that they've been giving the checklist to manufacturers but haven't been getting them back. Darrell Flocken has two examples, one for built-for-purpose and one for a not-built-for-purpose device. Jason Jordan from GIPSA said that they'd try it out. Doug Bliss and Jim Pettinato have volunteered to answer any questions that might arise as the labs attempt to use the checklist.

The Sector asked that the revised checklist continue to be used by the labs.

Discussion:
As we meet with each Sector jointly, we can get an updated report on the trial and decide if we're ready to recommend it for Pub. 14. We can also look at the language from D-31 in more detail in an effort to craft guidance in line with NCWM/NTEP philosophy.

This checklist was discussed during the NTEP lab meeting, and Darrell Flocken received two submissions. One response was very helpful, and the other one said that everything was N/A pertaining to their device, except for a bit regarding calculating the CRC and sealing. In general, the labs said that even when they hand the checklist out, they usually don't get it back. We're pushing the labs to be a bit more proactive.

MDMD has only one lab. All the labs have been given a copy of the checklist, but we're not sure whether their lab has found it helpful.

Conclusion:
Darrell Flocken will continue to be a point of contact if businesses or the labs have questions, but no one has yet contacted him in that regard.

Again, the benefit of a separate section of Pub. 14 for software is evident for this agenda item.

4. NTEP Application for Software and Software-based Devices

Source:
NTEP Software Sector

Background:
The purpose of initiating this item was to identify issues, requirements and processes for type approving Type U device applications. It was suggested that it may be useful to the labs to devise a separate submission form for software for Type U devices. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked-up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, “The approval applicant is responsible for the provision of all the required equipment and components.” This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4): This is the list of documents referred to in the checklist.

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.

Darrell Flocken and Jim Truex reviewed existing documentation required for obtaining certification in Pub. 14, administrative policy, and the application, to see what is already required. Administrative policy 9.1.7 was where this was found:

- Engineering specification
- Operating descriptions that characterize the type

NTEP evaluators already have the authority to request whatever documentation they need. We can provide them with a list of documents that we think would assist the evaluator in his job and also give the manufacturer a good idea of what they should be capable of providing.

Darrell Flocken suggested that this list could be added to administrative policy 9.1.7 in Pub. 14. Jim Truex suggested it could also be added to the application.

If we combine the two lists, it might appear as something like this:

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- A description of the user interface, communication interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.

- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.
- Engineering specification.
- Operating descriptions that characterize the type.

A statement could be made along the lines of, "If not included in the operating manual, provide the following, as applicable."

After the last sentence in 9.1.7, this could be added:
As part of the type evaluation submission, the following information should be provided for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.) , how to view it, and how it is tied to the software.**
- **An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.**

These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the labs is gathered. The Sector recommends including the above bulleted list as an introduction to the checklist as part of our recommendation to include the checklist from agenda item 3 in Pub. 14. As a description of the accuracy of the measuring algorithms, simply declaring the type and class being aimed for may be sufficient. This list should reflect the needs of the labs for an evaluation. The bulleted list and the paragraph before it should be brought to the labs for an initial review and their input.

There may be concerns with disclosure of proprietary information. Jim Truex says that the labs already protect other proprietary information. If the information provided is sufficiently high level, even theft of the data shouldn't cause too much of a concern.

While working on writing requirements for Pub. 14 from the checklist we've designed, we considered altering the second bullet point in our proposal for 9.17, so that it will require a description of how the software version or revision identifier is tied to the software itself.

At the 2016 meeting, it seemed that the goal of this agenda item has somewhat shifted back to the original purpose, which is how do we communicate to applicants the expectations related to software based devices? Diane Lee suggested we review the OIML requirements for documentation. The comment was made from the floor that OIML may go further than we are currently prepared to recommend. Jason Jordan expressed his opinion that moving forward with this item will be helpful for the labs. Darrell Flocken and Jim Truex think this should be added to the Application section. If limited to that section, it shouldn't require approval from any of the other Sectors. Doug Bliss suggested that it might be easier to provide examples that do not meet acceptable standards.

As we began discussing the training of field inspectors, Darrell Flocken asked that we also provide further training for lab inspectors. There's an annual lab meeting typically around April, in 2017 it will be in Annapolis, MD.

Discussion:
The Software Sector's recommendation is to add the requirements to the Application section. The Software Sector agreed to provide support for any desired training of lab personnel at the April meeting.

Jim Pettinato suggested that this agenda section has become largely redundant to the previous agenda section (the checklist). As time has passed, we've begun to address software the same regardless of its platform. Built-for-Purpose and Not-Built-for-Purpose differentiation seems less relevant. Doug Bliss pointed out that we still need to address how to communicate these issues to manufacturers. For now, we will continue with two different agenda items since the contents of the checklist are a separate issue from how we want to address / communicate the requirements.

As previously stated in earlier meetings, the labs can ask for any documentation they like, but it would be good to give manufacturers advance notice. Part of the Technical Policy (NTEP or individual codes) could include a requirement to fill out the checklist. Jim Truex suggested our best path forward may be to take the checklist (once we're sure it's mature) to the NTEP committee and ask them to add it to their policy.

Though we haven't thoroughly considered adding this to Hdbk. 44, Darrell Flocken pointed out that there is a portion of the handbook that pertains only to type evaluation.

Jim Truex's suggestion is probably the option with the best chance of success, but it will require some convincing. Doug Bliss suggested that we may need to put together a presentation like what we did for the adoption of our G-S.1 wording.

9.3 of Administrative Policy describes how to prepare for type evaluation. It might be better to add our suggested wording there instead of 9.1.7. Jim Pettinato found a page on NCWM's website that describes what's needed for a type evaluation. He suggested we could add our checklist to the list of documents there. The NTEP Committee decides what's posted on the website.

Jim Truex thinks we may need to come up with a list of software parameters and functions that are required to be protected. This will be a lot of work, but it may be the right answer, generating a separate section in Pub. 14 (and/or Hdbk. 44) pertaining specifically to software.

Darrell Flocken suggested we create a new agenda item for addressing the NTEP Committee. They meet 4 times a year. In fact, they meet 2 weeks from now (after the NUMA meeting) in Saratoga Springs, NY. Thereafter they meet in July.

Jim Truex said that he doesn't think that the software security concept has progressed far enough for it to be adopted in any formal manner.

The group discussed whether a list of sealable parameters should include device-specific parameters as well as software-specific parameters (e.g. CRC), or only the latter. The latter should be a fairly short list, including such parameters as:

- Replacing software
- Access to critical sections of the software

Historically, requirements for software-only applications haven't been as high as requirements for software applications that include hardware. The number of software-only applications has increased dramatically over the last few years.

The topic arose once again that we propose to the NTEP Committee we add a software specific section to Pub. 14. We may not know exactly what we want to include, but we could get the ball rolling by presenting a set of examples of situations that show the need. Jim Truex thinks that the NTEP Committee will ask whether this needs a change to Hdbk. 44. We need to address that in any sort of presentation we make to them. Dick Suiter suggested that we add a requirement to HB44 that the software be sealable, which is a bit of a difference from making changes to software evident. G-S.2. appears to address this in its mention of avoiding facilitation of fraud. The philosophy of sealing and method of sealing also cover this. We want to recommend adding a separate section to Pub. 14 for software, a list of sealable parameters, explain that going to the separate sectors isn't working, and explain that manufacturers will need to address both our software section as well as application-specific portions of Pub. 14.

Conclusion:
We will provide an outline for the proposed Pub 14 section prior to the meeting in two weeks, to run past the NTEP Committee to get their feedback. We want to make sure this is a viable approach, in their opinion.

5. Training of Field Inspectors

Source:
NTEP Software Sector

Background:
During discussions at the 2009 NTEP Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources

System Verification Tests
NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
   1.3 Software version/revision.(added at the 2017 Software Sector meeting)
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2

Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.
7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.

8. Over capacity.
8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]
8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.
9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)
10. Over capacity.
10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition, Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Co-Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices*.

The PDC is focused on training sessions at the moment, so it's unsure how much time they'd have to review this currently.

It was suggested by Jim Truex and Darrell Flocken we make it part of our report as an attachment or an appendix of the meeting minutes. Then we can send out an email notifying the Software Sector members as to where to find it.

Alternatively, we could forward the document to the PDC Committee, tell them it was our starting point, and ask them for their suggestions.

The Sector would like to continue exploring means by which it can be of assistance in training of field inspectors as software and electronic systems become more and more prevalent in their daily tasks.

It was also suggested we contact Ross Anderson, a paid consultant working with the PDC committee, to ask his opinion on how the Software Sector could best proceed to assist in the training of field inspectors. The Sector chair, Jim Pettinato, will act as primary point of contact for this communication.

For the Grain Analyzer Sector, Diane Lee thought it would take some time to put together some training material, as they do not currently have anything in place for software requirements.

Examples from completed checklists would be very helpful when putting together field inspector training. A lot of training videos have been recently generated. Doug Musick suggested that we recommend adding this to the agenda for the PDC Committee. Certification exams could be updated more easily, on a state-by-state level. It might be better to make software a separate exam.

Diane Lee suggested we look at developing a basic course for software, incorporating specific guidelines for specific device types.

Amanda Dubin was concerned about having the field inspectors know all the different existing software, which is a monumental task. Instead, the training should focus on how to find the pertinent CoC and look up information from it on the website. Ideally, down the road there could be some sort of database or software tool disseminated to field inspectors to assist in the look up of certificate numbers and the approved version number(s) for the software for a particular device, and even instructions on how to view/print the audit trail.

Jim Truex holds a meeting once a year for the lab evaluators. Darrell Flocken suggested that we also focus on training them on software. Diane Lee mentioned that NIST has been having manufacturers coming in to provide training on, for example, how to access the audit trail.

Discussion:

A video explaining the different sealing requirements was developed several years ago. It was intended for inspectors. NIST has given this video out at several training sessions.

The very first thing a field inspector needs to do is determine whether the software/system is metrological. Jim Truex said that they need guidance in figuring this out.

Inspectors are trained to look for a CC and look it up. A lot of the time this occurs during initial implementation of new equipment.

Not all devices are evaluated by NTEP, so they won't have a CC. That might be because NTEP hasn't established an evaluation for that type of device yet.

There are only 4 states that don't participate in NTEP. Two of those do participate informally (not legally required).

Conclusion:

Jim Pettinato suggested that at least for the short term, we work with California on an EPO.

6. Retrieval of Audit Log information

Source:
Adam Oldham, Gilbarco

Background/Discussion:
The current requirements for a Category III audit trail include printing of log on demand. However, many devices are approved standalone and can be connected to systems that are approved standalone. How could Category 3 audit trail mechanisms be approved in situations where multiple devices need to work together to attain it? How can a device maintain Category 2 and 3 approvals in this scenario? What alternatives to printing can be considered as potentially valid solutions? (files, laptop, flash drive, etc.).

This was discussed during the Measuring Sector's meeting on 9/15. The wording suggested was not agreed upon. Adam Oldham would like to have the Software Sector's suggestions, so he can put together a proposal for next year.

The US has rather unique requirements for printing the Category 3 audit trail, which are quite unwieldy – both in terms of the actual printing process (and results), as well as the needed approvals (the example provided by Adam Oldham required an approval for each and every POS system that might be connected to their system). The most similar is from Mexico, but they require an electronic copy.

Darrell Flocken reported that there has been a little movement forward – alternative methods are now allowable, to some degree, but it's dependent on what the states are going to allow, and it still requires the ability to print it. The change will be in LMD Code S.2.2., not in Handbook 44 G-S.2.2.

We discussed the difficulty of requiring that the electronic data be printable on-site, given that some sites don't have any printers, and other sites may have printers attached to computers that are restricted in what can be used to attach to them.

In Mexico, Gilbarco relies upon laptops being present, supplied by the auditing company.

LMD Pub. 14 has a section in Appendix B Requirements for Metrological Audit Trails on the event logger, and that information doesn't seem to be in Handbook 44. In fact, it may even contradict what's in the LMD Pub. 14. In practice, what's in Pub. 14 tends to be more influential with evaluators.

Adam Oldham volunteered to work on the wording for a proposal to present at the 2016 Software Sector meeting for review, but was not in attendance at the meeting, so the item was tabled.

The chair has anecdotal evidence that other parties have also expressed interest in specifying alternate methods for distributing audit trail information aside from the current 'printing' requirements. This discussion should be continued at the 2017 meeting.

Discussion:
In 2016 the Conference worked on some changes, but some states don't care for them. Previously only printouts were allowed, but now an "alternate method" is potentially allowed if all parties agree. Jim Truex thinks this issue has moved as far forward as it probably will for the time being.

Darrell Flocken suggested that if/when we get a separate software section added to Pub. 14, we ensure that our wording match the other similar sections in Pub. 14.

Conclusion:
This agenda item was closed by the Sector.

NEW ITEMS

7. Use of GPS Receivers and Mapping Software for Trade (e.g. fare determination)

Source: Software Sector

Background:

See the 2016 Software Sector Meeting Summary for additional background.

There were a few presentations at the Interim Meeting on this subject. The 2016 Annual Meeting archive (Denver 2016) includes a presentation from Lyft from that meeting.

Ambler Thompson has discussed this subject with European officials. One issue is traceability of the time stamp(s). You can also calculate velocity based upon the phase shift of the GPS signal, though it requires a high-end, survey-grade GPS receiver ($50k each). Car companies can use these devices to obtain a great deal of data.

Uber and Lyft claim that they are not billing upon GPS data, but rather a pre-negotiated contract based upon distance, time, and type of vehicle. Doug Bliss has been told that the bill is based upon the starting GPS location from the driver's phone, the ending GPS location from the same phone, and a calculation of the shortest distance from Google Maps. If the driver's phone doesn't have a great GPS receiver, or if the reception is bad so it's relying upon cell towers, etc., that's a problem. We're also not sure just how accurate Google Map's route calculation is. Also, Google Maps is a disinterested third party whose database is being used for a purpose they didn't specifically authorize.

Discussion:

Both Uber and Lyft provided presentations at the 2017 Interim Meeting to address some of the concerns that have been raised.

There is a US working group devoted to this subject now. There are three commercial parties – the driver, the rider, and the company itself. The driver is the one providing the phone that is relied upon for measuring the time and distance. There is an option for an up-front fare, which doesn't fall under W&M jurisdiction. The driver's compensation is based upon time and distance, so that is pertinent for W&M, as is the rider's cost if a destination isn't provided. Google Maps isn't being evaluated, just the Uber/Lyft software. The focus for testing is inputs and outputs. GPS data is traceable by NIST, which accounts for a different approach between the US and Europe. Google Maps is not traceable.

There are two proposals before the S&T Committee pertaining to these systems. One is for transportation management systems (i.e. Uber and Lyft), and if approved it would be on a test basis; systems wouldn't be red-tagged based upon it yet. The other one is for amending the existing taxi meter code to address these systems. Both will be voted upon in July. There's a third item, but it isn't as extensive of the other two items.

Conclusion:

At this stage, there isn't much for the Software Sector to do on this subject. Jim Pettinato asked that the members of the Software Sector review the proposals in Pub. 16 pertinent to this issue.

8. Next Meeting

Background:
The sector is on a yearly schedule for NTEP Software Sector Meetings. Now that we've adopted a joint meeting system, the next Sector joint meeting will coincide with one of the remaining Sector meetings.

Everyone thinks that the joint sessions have been productive; however, Darrell Flocken pointed out that if we do get a separate software section in Pub. 14 we may want to have a separate working session to work out wording/terminology before meeting with the other sectors for feedback.

If we stick with the same rotation, the Weighing Sector would be the next meeting, though that would a year and a half from now. The MDMD Work Group will meet again in May 2018, and they're asking that we meet with them again since they anticipate a lot of changes in their functionality in that time. There is less pressure to meet right before regional meetings because we aren't currently proposing changes to Hdbk. 44 though our proposal to the NTEP Committee might have to go through the regionals.

Conclusion:
After some consideration, we decided we would meet with the Weighing Sector in August 2018.

National Type Evaluation Program (NTEP)
Software Sector Meeting Summary

August 23rd, 2018 / Louisville, KY
(in conjunction with the Weighing Sector)

INTRODUCTION

The charge of the National Type Evaluation Program (NTEP) Software Sector is important in providing appropriate type evaluation criteria for software-based weighing or measuring device based on specifications, tolerances and technical requirements of *NIST Handbook 44* Section 1.10 General Code, Section 2 for weighing devices, Section 3 for liquid and vapor measuring devices, and Section 5 for taximeters, grain analyzers, and multiple dimension measuring devices. The sector's recommendations are presented to the NTEP Committee each January for approval and inclusion in *NCWM Publication 14 Technical Policy, Checklists, and Test Procedures* for national type evaluation.

The sector is also called upon occasionally for technical expertise in addressing difficult *NIST Handbook 44* issues on the agenda of the National Conference on Weights and Measures (NCWM) Specifications and Tolerances (S&T) Committee. Sector membership includes industry, NTEP laboratory representatives, technical advisors and the NTEP Administrator. Meetings are held annually, or as needed and are open to all NCWM members and other registered parties.

Suggested revisions are shown in **bold face print** by **~~striking out~~** information to be deleted and **underlining** information to be added. Requirements that are proposed to be non-retroactive are printed in ***bold faced italics***.

Table A
Table of Contents

Title of Content **Page**

Table B
Glossary of Acronyms and Terms

Acronym	Term	Acronym	Term
BIML	International Bureau of Legal Metrology	OIML	International Organization of Legal Metrology
CC	Certificate of Conformance	OWM	Office of Weights and Measures
EPO	Examination Procedure Outline	PDC	Professional Development Committee
NCWM	National Conference on Weights and Measures	S&T	Specifications and Tolerances Committee
NIST	National Institute of Standards and Technology	SMA	Scale Manufacturers Association
NTEP	National Type Evaluation Program	WELMEC	European Cooperation in Legal Metrology

Details of All Items
(In order by Reference Key)

WELCOME

Since the Software Sector meeting was a joint meeting with the Weighing Sector, some time was set aside to meet and greet both new and familiar faces.

STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

Attendees of the 2018 NCWM Interim and Annual Meetings were asked to share any relevant comments or discussion that took place during the open hearings or NCWM Standards and Tolerances (S&T) committee working sessions. Results related to items on our Agenda were of particular focus.

Dr. Katya Delak, NIST, Office of Weights and Measures (OWM), provided a synopsis of international activity that relates to the work of the sector. (See appendix B)

JOINT SESSION PROGRESS REPORT, ACTIVE ITEMS OF MUTUAL INTEREST

This is the second joint meeting of these Sectors. To make sure we make the most of the time a quick review of the agenda items from both Sectors will be held to identify those that require collaboration, so all participants have a solid foundation for discussion. As part of this review, items of particular importance or interest should be allocated more time during the joint session day.

SOFTWARE SECTOR PRESENTATION

The Software Sector Technical Advisor gave a brief presentation outlining the problems the Sector had been asked to consider and some of the consensus that has been reached to date.

CARRY-OVER ITEMS

1. Software Identification / Markings

Source:
NTEP Software Sector

Background:
See the 2017 Software Sector Meeting Summary for more background on this item.

Since its inception, the sector has wrestled with the issue of software identification and marking requirements. Numerous changes to the HB44 language were attempted and though support for the concepts was expressed, resistance to specific language made the course difficult. Finally, in 2015 in a joint meeting with the Measuring Sector, some additional fine tuning on the recommended changes to G-S.1 was done and we felt we had addressed everyone's concerns and had language ready to be voted upon for adoption. The recommended language is below.

Amend *NIST Handbook 44:* G-S.1. Identification as follows:

G-S.1. Identification. – All equipment, except weights and separate parts necessary to the measurement process but not having any metrological effect**,** shall be clearly and permanently marked for the purposes of identification with the following information:

(a) the name, initials, or trademark of the manufacturer or distributor;

(b) a model identifier that positively identifies the pattern or design of the device;

(1) The model identifier shall be prefaced by the word "Model," "Type," or "Pattern." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). The abbreviation for the word "Model" shall be "Mod" or "Mod." Prefix lettering may be initial capitals, all capitals, or all lowercase.
[Nonretroactive as of January 1, 2003]
(Added 2000) (Amended 2001)

(*c*) *a nonrepetitive serial number, except for equipment with no moving or electronic component parts and* ***~~not-built-for-purpose software-based software devices~~*** ***software****;*
[Nonretroactive as of January 1, 1968]
(Amended 2003)

(1) The serial number shall be prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required serial number.
[Nonretroactive as of January 1, 1986]

(2) Abbreviations for the word "Serial" shall, as a minimum, begin with the letter "S," and abbreviations for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., S/N, SN, Ser. No., and S. No.).
[Nonretroactive as of January 1, 2001]

(*d*) the current software version or revision identifier for not-built-for-purpose software-based devices~~;~~ **manufactured as of January 1, 2004 and all software-based devices or equipment manufactured as of January 1, 2022;**
~~[Nonretroactive as of January 1, 2004]~~
(Added 2003) **(Amended 2017)**

(1) The version or revision identifier shall be:

i. *prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;*
[Nonretroactive as of January 1, 2007]
(Added 2006)

Note: If the equipment is capable of displaying the version or revision identifier but is unable to meet the formatting requirement, through the NTEP type evaluation process, other options may be deemed acceptable and described in the CC.
(Added 2017)

ii. ***continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.***
[Nonretroactive as of January 1, 2022]
(Added 2017)

(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). ***Prefix lettering may be initial capitals, all capitals, or all lowercase.***
[Nonretroactive as of January 1, 2007]
(Added 2006) (Amended 2017)

(e) an National Type Evaluation Program (NTEP) Certificate of Conformance (CC) *number or a corresponding CC Addendum Number for devices that have a CC.*

(1) The CC Number or a corresponding CC Addendum Number shall be prefaced by the terms "NTEP CC," "CC," or "Approval." These terms may be followed by the word "Number" or an abbreviation of that word. The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.)
[Nonretroactive as of January 1, 2003]

The required information shall be so located that it is readily observable without the necessity of the disassembly of a part requiring the use of any means separate from the device. (Amended 1985, 1991, 1999, 2000, 2001, 2003, **~~and,~~** 2006 **and 2017**)

The amended proposal was Accepted as a Voting item at the 2016 Interim meeting and passed at the 2016 Annual Meeting.

Since future work on this item depends on the expiration of the window for compliance (2022), the Sector agreed to table this item until 2020/2021, when we can again begin to discuss further modifications with the eventual goal of eliminating G-S.1.1 and the differentiation between built-for-purpose and not-built-for-purpose.

In July of 2016 the MDMD Work Group addressed some of these issues pertaining to software running on small devices such as phones that have very small screens. They discussed prioritization of what needed to be displayed, such as CC so that the remainder of the information can be looked up.

Discussion:
The group estimated the scope of work remaining and decided it is not necessary to start working on G-S.1 yet.

Conclusion:
This agenda item remains tabled until 2020.

2. Identification of Certified Software

Source:
NTEP Software Sector

Background:
See the 2017 Software Sector Meeting Summary for more background on this item.

This item originated as an attempt to answer the question "How does the field inspector know that the software running in the device is the same software evaluated and approved by the lab?"

In 2010, the sector recommended the following change to *NIST Handbook 44*, General Code: G-S.1(d) to add a new subsection (3):

(d) the current software version or revision identifier) the current software version or revision identifier for not-built-for-purpose software-based devices manufactured as of January 1, 2004 and all software-based devices or equipment manufactured as of January 1, 2022;
(Added 2003) (Amended 2016)

(1) The version or revision identifier shall be:

i. prefaced by words, an abbreviation, or a symbol, that clearly identifies the number as the required version or revision;
[Nonretroactive as of January 1, 2007]
(Added 2006)

***Note**: If the equipment is capable of displaying the version or revision identifier but is unable to meet the formatting requirement, through the NTEP type evaluation process, other options may be deemed acceptable and described in the CC.*
(Added 2016)

ii. continuously displayed or be accessible via the display. Instructions for displaying the version or revision identifier shall be described in the CC. As an alternative, permanently marking the version or revision identifier shall be acceptable providing the device does not always have an integral interface to communicate the version or revision identifier.
[Nonretroactive as of January 1, 2022]
(Added 2017)

*(2) Abbreviations for the word "Version" shall, as a minimum, begin with the letter "V" and may be followed by the word "Number." Abbreviations for the word "Revision" shall, as a minimum, begin with the letter "R" and may be followed by the word "Number." The abbreviation for the word "Number" shall, as a minimum, begin with the letter "N" (e.g., No or No.). Prefix lettering may be initial capitals, all capitals, or all lowerc**ase.***
[Nonretroactive as of January 1, 2007]
(Added 2006) (Amended 2017)

(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Also the sector recommended the following information be added to *NCWM Publication 14* as explanation/examples:

- Unique identifier must be displayable/printable on command or during operation, etc.

- At a minimum, a version/revision indication (1.02.09, rev 3.0 a, etc). Could also consist of / contain checksum, etc. (crc32, for example)

This original item was eventually withdrawn, and the proposal was split into two separate items. The critical need to include version/revision in the marking requirements for all software-based devices was pushed forward and passed independently.

In addition, the sector considered the following information to be added to *NCWM Publication 14* as explanation/examples:

- The current software identifier must be displayable/printable on command during operation (or made evident by other means deemed acceptable by G-S.1.)
- At a minimum, the software identifier must include a version/revision indication (1.02.09, rev 3.0 a, etc). It could also consist of / contain checksum, etc. (crc32, for example).
- The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.

Other questions previously brought up that have not really been satisfied to date are:

- If we allow hard-marking of the software identifier (the sector has wavered on this in the past), does the above wording then imply that some mechanical means is required (i.e. physical seal) to “inseparably link” the identifier to the software?
- If a device is capable of doing so, does it still have to be able to display, print or communicate the identifier somehow, even if it is hard-marked?

Regarding field inspection and locating the required information: The list of acceptable menu text and symbols in Appendix A are intended to assist the labs in finding the certification number. The sector noticed no action by the sectors had been taken when this list was circulated for comment. We would like to remind them that we would like to have it reviewed. We feel that this belongs in, for example, the Weighing Device Pub. 14, page DES-22, Section 3; the Belt – Conveyor Scales, page BCS-10, Section 8.7; the Measuring Devices, page LMD-21, Section 1.6; the Grain Moisture Meter, page GMM-14, Section 1 (G.S.1); and Near Infrared Grain Analyzers, page NIR-8, Section 1 (G.S.1).

Tina Butcher mentioned that the Weighing Sector has a Weighing Checklist that has a similar set of approved symbols, so the examples shown in Appendix A would be in line with their current practice.

Since the recommended new G-S.1 language was voted on and adopted in 2016, we can now move forward on this item and consider adding to *NCWM Publication 14* the specifics that we have been discussing related to presenting the software identification.

Darrell Flocken asked whether it’s a specification or information. That would determine whether it should belong in HB44 or only in Pub. 14. One possibility is below:

(3) The version or revision identifier shall be directly and inseparably linked to the software itself.

Note: The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Concern was expressed that this could cause confusion with field inspectors. Software separation isn’t something that’s intended to be useful in the field, it is intended to ease type approval and software maintenance release processing. - This would lend weight to the argument of keeping it in Pub. 14.

If the Sector desires to include this in Pub. 14, we would need to identify all the sections where this concept would need to be added. The Software Sector doesn't have the authority to add it to the other sectors' Pub. 14's. Darrell Flocken reported that a note regarding the concept of software separation has already been added to several of the various Pub. 14 sections.

The Chair proposed that we table Agenda Item 2 until 2021, and that we continue to pursue implementing the checklist in Pub. 14. Darrell Flocken suggested that the Software Sector recommend that the various sectors adopt this for their Pub. 14's. It would take a year or so, to make it through all the various sectors. A note could be added saying that a device can't be rejected if it doesn't meet this requirement in the checklist until 2022. It was agreed that we would table this item until the 2021 meeting, at which time we will propose the following (updated) wording for the 2022 Pub. 14:

3. Additional Marking Requirements- Software

Identification of Certified Software:
The manufacturer must describe and possibly demonstrate how the version or revision identifier is directly and inseparably linked to the metrologically significant software. Where the version revision identifier is comprised of more than one part, the manufacturer shall describe which portion represents the metrological significant software and which does not.

Note: Manufacturers may choose to separate metrologically significant software from non-metrologically significant software. Separation would allow the revision of the non-metrological portion without the need for further evaluation. In addition, non-metrologically significant software may be updated on devices without breaking a seal, if so designed. Separation of software requires that 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). If the separation of the software is not possible or needed, then the software is metrologically significant as a whole.

At the 2017 joint meeting, the MDMD Work Group discussed adding the section regarding linking of identifier to the software to their section in Pub. 14. There were no objections, so Darrell Flocken said he'd add it for next year's publication. A note shall be added that this is voluntary until 2022.

Also, we further discussed the idea of software separation, especially in how it pertains to the difference between the terms "metrologically significant" and "legally relevant". Some legal requirements have nothing to do with metrology. There is a difference in how the US regards this (since each state can have different legal requirements) vs. the philosophy in Europe. There isn't a definition of "metrologically significant" in Handbook 44, but Publication 14 has a description of all the parameters that needs to be sealed, which includes both metrologically significant and legally relevant parameters.

A definition of "metrologically significant" could be helpful, but Darrell Flocken suggested that we make sure it doesn't contradict VCAP's administrative policies.

Handbook. 44 does contain a definition for "metrological integrity".

Type evaluation is the time at which decisions are made regarding which exact parameters are sealable. According to Jim Truex, the US has never been able to come to a consensus on this subject.

Jim Pettinato suggested that we work offline to generate a description intended to provide guidance on what we mean by "metrologically significant". Jim Pettinato, Doug Bliss, Dr. Ambler Thompson, and Kevin Detert volunteered to make up a subcommittee to address this subject.

We also considered the issue of having to adopt a general software requirement to multiple sections of Publication 14 to address essentially the same requirement for each category of device separately. The idea was floated by the

Sector that perhaps a new section should be added to Publication 14 specific to software that applies to all metrologically significant software in all devices types that might contain such. Rather than formally suggesting this be done, we decided to informally run the idea past the Specifications and Tolerances committee. That way, if there was little interest or strong objection, we wouldn't waste time generating a draft.

How the Sector decides to progress on this item is dependent on the Board's decision regarding a separate section on software for Publication 14. If the decision is to grant the Sector's wishes, then we would start crafting language for our new Section. Otherwise, we can consider the suggested language put forth in the last meeting.

Discussion:
If the Software Sector gets its own section in Publication 14, we may not need to alter HB44 regarding this specific agenda item, according to Darrell Flocken. There is a general NTEP technical policy within Pub. 14, which may be the best place to address communicating the requirements for evaluation of software and software-based devices and the need to include type compliant software version/revision information on the certificate of conformance.

Conclusion:
This agenda item remains tabled until a decision on the direction for Publication 14 is made by the NTEP committee.

3. Software Protection / Security

Source:
NTEP Software Sector

Background:
See the 2017 Software Sector Summary for additional background on this item.

The Sector continued to develop a proposed checklist for *NCWM Publication 14*. The numbering will still need to be added. This is based roughly on R 76 – 2 checklist and discussions beginning as early as the October 2007 NTEP Software Sector Meeting. 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 laboratories agreed to use this check list on one of the next devices they have in the lab and report back to the sector on what the problems may be. In February 2011, the North Carolina laboratory was also given a copy of the check list to try.

The labs using this checklist on a trial basis indicated that there was some confusion as to versions/wording. There may be more than one version in circulation. The version shown in this Summary shall be used henceforth.

The checklist as updated during the 2014 meeting:

1. Devices with Software

1.1. Declaration of the manufacturer that the software is used in a fixed hardware and software environment. **The manufacturer should indicate whether it's solely software or includes hardware in the system. Can the software be changed after the system has been shipped without breaking a seal?** AND □ Yes □ No □ N/A

1.2. Cannot be modified or uploaded by any means after securing/verification. **With the seal intact, can you change the software?** □ Yes □ No □ N/A

Note: It is acceptable to break the "seal" and load new software, audit trail is also a sufficient seal.

1.3. The software documentation contains:

1.3.1. Description of all functions, designating those that are considered metrologically significant. □ Yes □ No □ N/A

1.3.2. Description of the securing means (evidence of an intervention). □ Yes □ No □ N/A

1.3.3. Software Identification, including version/revision. **It may also include things like name, part number, CRC, etc.** □ Yes □ No □ N/A

1.3.4. Description how to check the actual software identification. □ Yes □ No □ N/A

1.4. The software identification is:

1.4.1. Clearly assigned to the metrologically significant software and functions. □ Yes □ No □ N/A

1.4.2. Provided by the device as documented. □ Yes □ No □ N/A

1.4.3. Directly linked to the software itself. **This means that you can't easily change the software without changing the software identifier. For example, the version identifier can't be in a text file that's easily editable, or in a variable that the user can edit.** □ Yes □ No □ N/A

2. Programmable or Loadable Metrologically Significant Software

2.1. The metrologically significant software is:

2.1.1. Documented with all relevant (see below for list of documents) information. *The list of docs referred to exists in agenda item 5.* □ Yes □ No □ N/A

2.1.2. Protected against accidental or intentional changes. □ Yes □ No □ N/A

2.2. Evidence of intervention (such as, changes, uploads, circumvention) is available until the next verification / inspection (e.g., physical seal, Checksum, Cyclical Redundancy Check (CRC), audit trail, etc. means of security). □ Yes □ No □ N/A

3. Software with no access to the operating system and/or programs possible for the user. **This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.**

3.3. 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

3.4. Check whether the manufacturer has submitted a written declaration of the completeness of the set of commands. □ Yes □ No □ N/A

4. Operating System and / or Program(s) Accessible for the User. **Complete this section only if you replied No to 1.1.**

4.5. 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 Weights and Measures jurisdiction and type-specific parameters). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

4.6. 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). **This is a declaration or explanation by the manufacturer.** □ Yes □ No □ N/A

4.7. Check whether the manufacturer has provided a description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.

4.8. Check that there is guidance related to the software identification (version, revision, etc.), how to view it, and how it is tied to the software.

4.9. Check that the manufacturer has provided an overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.

5. Software Interface(s)

5.10. Verify the manufacturer has documented:

5.10.1. **If software separation is employed, t**he program modules of the metrologically significant software are defined and separated. □ Yes □ No □ N/A

5.10.2. **For software that can access the operating system or if the program is accessible to the user, t**he protective software interface itself is part of the metrologically significant software. □ Yes □ No □ N/A

5.10.3. The functions of the metrologically significant software that can be accessed ~~via the protective software interface~~. □ Yes □ No □ N/A

5.10.4. The **metrologically significant** parameters that may be exchanged ~~**via the protective software interface**~~ are defined. □ Yes □ No □ N/A

5.10.5. The description of the functions and parameters are conclusive and complete. □ Yes □ No □ N/A

5.10.6. There are software interface instructions for the third party (external) application programmer. □ Yes □ No □ N/A

This checklist was discussed during the 2017 NTEP lab meeting, and Darrell Flocken received two submissions. One response was very helpful, and the other one said that everything was N/A pertaining to their device, except for a bit regarding calculating the CRC and sealing. In general, the labs said that even when they hand the checklist out, they usually don't get it back. We're pushing the labs to be a bit more proactive.

MDMD has only one lab. All the labs have been given a copy of the checklist, but we're not sure whether their lab has found it helpful.

Again, the benefit of a separate section of Pub. 14 for software is evident for this agenda item.

Discussion:
Darrell Flocken shared (anonymously) some results from the NTEP labs. There were three checklists returned over the last year. One submission included commentary from the company responding to the checklist regarding the difference between embedded systems versus open systems. That submitter used the WELMEC guidelines.

Darrell Flocken reported that, in general, it seems that companies are starting to respond more thoughtfully to the checklist. In prior years, it seemed like they'd simply just checked everything off.

There appears to be a gap between the companies responding to the checklist and the NTEP labs perceiving use in the responses. There's a need for an explanation of what responses to the various questions mean to the NTEP lab inspectors, which should be in plain language, similar to the 2014 presentation on the general concepts of the Software Sector's work.

We also discussed the need to formalize how the checklist is distributed. Cardinal reported that they hadn't received it as part of a type approval application packet, and it seems they're not unique.

It was mentioned that Mexico now considers many things "software", including PAL's, GAL's, etc. At one time we tried to craft our own definition of software without much luck. We may be able to reference an international definition.

The VCAP program should reference the software identifier and version/revision, but until NTEP is consistent on how the software identifier and version/revision is recorded on the CC, this isn't feasible. VCAP was originally intended as an assessment whether a particular implementation meets type.

Conclusion:
Darrell Flocken will work on formalizing how the checklist is distributed. We will also need to work on crafting an explanation for the NTEP labs as to how the answers to the checklist benefit their inspectors.

4. NTEP Application for Software and Software-based Devices

Source:
NTEP Software Sector

Background:
The purpose of initiating this item was to identify issues, requirements and processes for type approving device applications, specifically for not-built-for-purpose software since it is now explicitly allowed. It was suggested that it may be useful to the labs to devise a separate submission form for software for these applications. What gets submitted? What requirements and mechanisms for submission should be available? Validation in the laboratories - all required subsystems shall be included to be able to simulate the system as installed.

Mr. Roach, California Division of Measurement Standards, stated that if the software package being evaluated supports platforms/subsystems from multiple manufacturers, testing should be done using at least two platforms/subsystems. Scale laboratories and scale manufacturers indicated that this is not usually done for scale evaluations.

Since the NTEP Committee passed the related item at NCWM Annual Meeting we will continue to work on this. Mr. Truex, NTEP Administrator, indicated that we can move in this direction, but felt that it was somewhat premature to develop this thoroughly now. At the point where the sector has developed checklist requirements, then we could move to perhaps add a subsection to current NTEP applications for applicable software. Refer to D-31.6.1. It was also agreed that there seems to be no reason for limiting the scope of this item to software-only applications, and hence all software/software-based devices could benefit from an enhanced application process. Hence the description of this agenda item was modified as shown in the marked-up heading.

Comments given at the meeting indicate that current practice does not require anything different for software / software based devices compared to any other type approval. It was also noted that for international applications, OIML D-31.6.5 states, “The approval applicant is responsible for the provision of all the required equipment and components.” This would likely also be the policy of NTEP.

Since the checklist is still being tried out by some of the laboratories, the sector is not quite ready to develop this fully. Some documentation that eventually might be required by applicants could include (from WELMEC doc. 7-2 Issue 4): This is the list of documents referred to in the checklist.

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.

Darrell Flocken and Jim Truex reviewed existing documentation required for obtaining certification in Pub. 14, administrative policy, and the application, to see what is already required. Administrative policy 9.1.7 was where this was found:

- Engineering specification
- Operating descriptions that characterize the type

NTEP evaluators already have the authority to request whatever documentation they need. We can provide them with a list of documents that we think would assist the evaluator in his job and also give the manufacturer a good idea of what they should be capable of providing.

Darrell Flocken suggested that this list could be added to administrative policy 9.1.7 in Pub. 14. Jim Truex suggested it could also be added to the application.

If we combine the two lists, it might appear as something like this:

- A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.
- A description of the user interface, communication interface, menus, and dialogs.
- The software identification (version, revision, etc.) and how to view it.
- An overview of the system hardware, e.g. topology block diagram, type of computer(s), type of network, etc, if not described in the operating manual.
- An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.
- The operating manual.
- Engineering specification.
- Operating descriptions that characterize the type.

A statement could be made along the lines of, "If not included in the operating manual, provide the following, as applicable."

After the last sentence in 9.1.7, this could be added:

As part of the type evaluation submission, the following information should be provided for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.) , how to view it, and how it is tied to the software.**
- **An overview of the security aspects of the operating system, e.g. protection, user accounts, privileges, etc.**

These documentation requirements will be considered as input for requirements that will eventually appear in *NCWM Publication 14* and the application paperwork. Further work by the sector to develop the *NCWM Publication 14* requirements is needed, after more input from the labs is gathered. The Sector recommends including the above bulleted list as an introduction to the checklist as part of our recommendation to include the checklist from agenda item 3 in Pub. 14. As a description of the accuracy of the measuring algorithms, simply declaring the type and class being aimed for may be sufficient. This list should reflect the needs of the labs for an evaluation. The bulleted list and the paragraph before it should be brought to the labs for an initial review and their input.

There may be concerns with disclosure of proprietary information. Jim Truex says that the labs already protect other proprietary information. If the information provided is sufficiently high level, even theft of the data shouldn't cause too much of a concern.

While working on writing requirements for Pub. 14 from the checklist we've designed, we considered altering the second bullet point in our proposal for 9.17, so that it will require a description of how the software version or revision identifier is tied to the software itself.

At the 2016 meeting, it seemed that the goal of this agenda item has somewhat shifted back to the original purpose, which is how do we communicate to applicants the expectations related to software based devices? Diane Lee suggested we review the OIML requirements for documentation. The comment was made from the floor that OIML may go further than we are currently prepared to recommend. Jason Jordan expressed his opinion that moving forward with this item will be helpful for the labs. Darrell Flocken and Jim Truex think this should be added to the Application section. If limited to that section, it shouldn't require approval from any of the other Sectors. Doug Bliss suggested that it might be easier to provide examples that do not meet acceptable standards.

9.3 of Administrative Policy describes how to prepare for type evaluation. It might be better to add our suggested wording there instead of 9.1.7. Jim Pettinato found a page on NCWM's website that describes what's needed for a type evaluation. He suggested we could add our checklist to the list of documents there. The NTEP Committee decides what's posted on the website.

Jim Truex thinks we may need to come up with a list of software parameters and functions that are required to be protected. This will be a lot of work, but it may be the right answer, generating a separate section in Pub. 14 (and/or Hdbk. 44) pertaining specifically to software.

The group discussed whether a list of sealable parameters should include device-specific parameters as well as software-specific parameters (e.g. CRC), or only the latter. The latter should be a fairly short list, including such parameters as:

- Replacing software
- Access to critical sections of the software

Historically, requirements for software-only applications haven't been as high as requirements for software applications that include hardware. The number of software-only applications has increased dramatically over the last few years.

The topic arose once again that we propose to the NTEP Committee we add a software specific section to Pub. 14. We may not know exactly what we want to include, but we could get the ball rolling by presenting a set of examples of situations that show the need. Jim Truex thinks that the NTEP Committee will ask whether this needs a change to Hdbk. 44. We need to address that in any sort of presentation we make to them. Dick Suiter suggested that we add a requirement to HB44 that the software be sealable, which is a bit of a difference from making changes to software evident. G-S.2. appears to address this in its mention of avoiding facilitation of fraud. The philosophy of sealing and method of sealing also cover this. We want to recommend adding a separate section to Pub. 14 for software, a list of sealable parameters, explain that going to the separate sectors isn't working, and explain that manufacturers will need to address both our software section as well as application-specific portions of Pub. 14.

We provided an outline for the proposed Pub 14 section prior to the NTEP committee meeting in two weeks, to gauge their opinion as to whether this is a viable approach. No action was taken until this year's Annual meeting, where the new NTEP committee chair guaranteed he would make it a priority to make progress on the proposal.

Discussion:
Darrell Flocken is trying to get an invitation for the Software Sector to the NTEP labs meeting in April, to be able to answer any of their questions and have a discussion on how software could be addressed more formally in submissions from applicants, and how the Sector can support the labs in their evaluations.

We need to provide a recommendation for an administrative change for the NTEP Committee's approval, via Darrell Flocken and Jim Truex. Since this would be a recommendation related to the policy, not the device code, it simplifies the process.

If the Software Sector does get its own section within Publication 14, the text may gain more notice if it's within that section rather than the general administrative policy; however, if it's within the general administrative policy, it wouldn't be hard to move it to the Software Sector's section of Pub. 14.

Conclusion:
The Software Sector recommends that this text be added as part of the existing 9.1.7 in Pub. 14 Administrative Policy:

Additionally, for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.), how to view it, and how it is tied to the software.**
- **An overview of the security aspects of the software(s), e.g. protection, user accounts, privileges, platforms, etc.**

Jim Pettinato will craft the formal proposal. Darrell Flocken asked the NTEP lab evaluators in attendance what they need from the Software Sector to help them interpret the documentation they will receive from the manufacturers in response to this requirement.

5. Training of Field Inspectors

Source:
NTEP Software Sector

Background:
During discussions at the 2009 NTEP Software Sector Meeting, the sector concluded that a new agenda item should be initiated specific to the training of field inspectors in relation to evaluating/validating software-based devices.

California has an Examination Procedure Outline (EPO) that begins to address this. Use *California Handbook 112* as a pattern template for how it could read.

Items to be addressed:

- Certificate of Conformance
- Terminology (as related to software) beyond what is in *NIST Handbook 44*.
- Reference materials / information sources

System Verification Tests
NOTE: Item numbers 1 through 5 apply to both weighing and measuring devices. Numbers 6 and 7 are specific to weighing devices; while numbers 9 and 10 apply to measuring devices.

1. Identification. The identification (ID) tag may be on the back room computer server and could be viewed on an identification screen on the computer monitor. The ID information may be displayed on a menu or identification screen. Though currently discouraged, some systems may be designed so the system must be shut down and reset to view the ID information. G-S.1 (1.10)
   1.1. Manufacturer.
   1.2. Model designation.
2. Provisions for sealing. G-S.8 [1.10]; S.1.11 [2.20]; S.2.2 [3.30]
   2.1. Verify sealing category of device (refer to Certificate of Approval for that system).
   2.2. Verify compliance with certificate.
3. Units of measure.
   3.1. A computer and printer interfaced to a digital indicator shall print all metrological values, intended to be the same, identically. G-S.5.2.2(a); G-S.5.1 [1.10]
   3.2. The unit of measure, such as lb, kg, oz, gal, qts, liters, or whatever is used, must agree.
4. Operational controls, indications and features (buttons and switches). Verify that application criteria and performance criteria are met (refer to Certificate of Approval).
   4.1. Any indication, operation, function or condition must not be represented in a manner that interferes with the interpretation of the indicated or printed values.
5. Indications and displays.
   5.1. Attempt to print a ticket. The recorded information must be accurate or the software must not process and print a ticket with erroneous data interpreted as a measured amount.

Weighing Devices

6. Motion detection.
   6.1. For railway track, livestock, and vehicle scales apply or remove a test load of at least 15d while simultaneously operating a print button, push-button tare or push-button zero. A good way to do this is to try to print a ticket while pulling the weight truck or another vehicle onto the scale. Recorded values shall not differ from the static display by more than 3d. Perform the test at 10%, 50% and 100% of the maximum applied test load. S.2.5.1(a) [2.20]; EPO NO. 2-3, 2.4
   6.2. For all other scales, apply or remove at least 5d. Printed weight values must agree with the static weight within 1d and must exactly agree with other indications. S.2.5.4(b) [2.20]; EPO NO. 2-3, 2.4
7. Behind zero indication.
   7.1 Apply a load in excess of the automatic zero setting mechanism (AZSM) and zero the scale. S.2.1.3 [2.20]; EPO NO. 2-3, 2.4, 2.5.2
   Example: On a vehicle scale have someone stand on the scale, then zero them off (AZSM is 3d). Remove the weight (person) and note the behind zero display (usually a minus weight value) or error condition.

7.2. Attempt to print a ticket. With a behind zero condition, (manually or mechanically operated) a negative number must not be printed as a positive value.

8. Over capacity.
   8.1. Manually enter a gross weight if permissible or apply a test load in excess of 105% of the scale's capacity. S.1.7 [2.20]; S.1.12, UR.3.9 [2.20]
   8.2. Attempt to print a weight ticket. A system must not print a ticket if the manually entered weight or load exceeds 105% of the scale capacity.

Measuring Devices

9. Motion detection.
   9.1. Initiate flow through the measuring element. Attempt to print a ticket while the product is flowing through the measuring chamber. The device must not print while the indication is not stable. S.2.4.1. (3.30)
10. Over capacity.
   10.1. Attempt to print a ticket in excess of the indicated capacity. A system must not print a ticket if the device is manually or mechanically operated in excess of the indicated value.

NOTE: Be aware of error codes on the indicator which may be interrupted as measured values.

Mr. Jordan, California Division of Measurement Standards, is already doing something similar, and he may be able to assist. Mr. Roach, California Division of Measurement Standards, will talk to him to see whether they're available. In addition, Mr. Parks, California Division of Measurement Standards, is based in Sacramento and a potential resource. If the meeting is held in Sacramento next year, they may be able to attend.

Mr. Truex, NTEP Administrator, pointed out that the PDC would also be a valuable resource on this subject. Mr. Pettinato, Co-Chair, will contact them.

**NIST Handbook 112- Examination Procedure Outline for Commercial Weighing and Measuring Devices.*

The PDC is focused on training sessions at the moment, so it's unsure how much time they'd have to review this currently.

It was suggested by Jim Truex and Darrell Flocken we make it part of our report as an attachment or an appendix of the meeting minutes. Then we can send out an email notifying the Software Sector members as to where to find it.

Alternatively, we could forward the document to the PDC Committee, tell them it was our starting point, and ask them for their suggestions.

The Sector would like to continue exploring means by which it can be of assistance in training of field inspectors as software and electronic systems become more and more prevalent in their daily tasks.

It was also suggested we contact Ross Anderson, a paid consultant working with the PDC committee, to ask his opinion on how the Software Sector could best proceed to assist in the training of field inspectors. The Sector chair, Jim Pettinato, will act as primary point of contact for this communication.

Jim Pettinato will contact Ross Anderson regarding the PDC Committee, offering the Software Sector's assistance in continuing to develop training pertaining to software.

Discussion:

Jim Pettinato is now a member of the PDC (Professional Development Committee), so he will be able to pass on any suggestions we may make. The PDC is making an effort to provide training modules/videos accessible to anyone, so everyone is on the same page. Darrell Flocken suggested that as these training modules are updated, we should provide relevant input.

There is a national EPO from NIST Office of W&M, HB112. Darrell Flocken recommended that we approach NIST regarding adding text regarding software. There are not EPO's for every equipment type. Rick said that HB112 is updated every year.

Darrell Flocken suggested that we attend the regional meetings to gain feedback on the sort of guidance the field inspectors need.

Rick said that the most value to the field inspectors would be to identify for them different means that software can be used to manipulate the metrological system. In particular, how can someone attempt to cheat using software?

Doug Bliss quickly reviewed HB112 and reported that the majority of it has to do with safety guidelines. Rick said that there are numerous references to HB44, which pertain more to the requirements for the inspections. HB112 has appendices that include step-by-step procedures. We may want to consider crafting our own procedure for a new appendix.

Adam mentioned that Mexico and Brazil (and China, to an extent) have a requirement for manufacturers to supply an auditing document when they submit for a type approval. This would be a big change for NCWM.

Conclusion:
It was suggested that perhaps a presentation on this subject at the main and regional NCWM meetings might be a good starting point. Jim Pettinato suggested an entry in the NCWM newsletter, targeted to inspectors, would also help. The newsletter is submitted quarterly. Darrell Flocken confirmed that submissions for the next newsletter are due January 15th. A helpful newsletter article could describe how to find the CC for a system that includes software. Brian Duncan volunteered to write a first draft.

Jim Pettinato suggested that members of the Software Sector download and review HB112, so that we can have a better idea regarding where we might best target additions to the text. We could have an online meeting to discuss and update the agenda prior to our next Software Sector meeting. Darrell Flocken or Jim Pettinato can set up an online meeting, which may be in late September.

Recommendations for changes to HB112 should go to Tina Butcher.

6. Use of GPS Receivers and Mapping Software for Trade (e.g. fare determination)

Source: Software Sector

Background:
There were a few presentations at the Interim Meeting on this subject. The 2016 Annual Meeting archive (Denver 2016) includes a presentation from Lyft from that meeting.

Ambler Thompson has discussed this subject with European officials. One issue is traceability of the time stamp(s). You can also calculate velocity based upon the phase shift of the GPS signal, though it requires a high-end, survey-grade GPS receiver ($50k each). Car companies can use these devices to obtain a great deal of data.

Uber and Lyft claim that they are not billing upon GPS data, but rather a pre-negotiated contract based upon distance, time, and type of vehicle. Doug Bliss has been told that the bill is based upon the starting GPS location from the driver's phone, the ending GPS location from the same phone, and a calculation of the shortest distance from Google Maps. If the driver's phone doesn't have a great GPS receiver, or if the reception is bad so it's relying upon cell towers, etc., that's a problem. We're also not sure just how accurate Google Map's route calculation is. Also, Google Maps is a disinterested third party whose database is being used for a purpose they didn't specifically authorize.

Doug Musick reported that the Uber contract is based upon a unit price, though they do provide an estimate to the customer.

At the 2017 meeting, it was determined that at this stage there isn't much for the Software Sector to do on this subject. Jim Pettinato asked that the members of the Software Sector review the proposals in Pub. 16 pertinent to this issue.

Discussion:
Katya Delak said that OIML may attempt to address this issue as well, probably within the next few years.

The 2018 Taxi Meter code has been changed, and approvals are not being generated for GPS-based technology. OBDC-based systems have been accepted for type approval. There are no Pub. 14 guidelines.

Conclusion:
As in 2017, it doesn't seem that there is anything the Software Sector needs to address on this subject currently. After some discussion, the members of the Software Sector agree that this agenda item can be removed from future meetings.

NEW ITEMS

7. New Publication 14 Section specific to Software

Background:
In the last few meetings, it has been recognized that there is significant difficulty aligning the various Sectors to maintain continuity and agreement in what changes go into each Sector's section of Publication 14. It also impedes the progress the Software Sector can make as we have to explain/defend our positions multiple times to different audiences. Hence, it was proposed while working on several of the carry-over items that a better process might be to segregate the software-specific requirements for type evaluation into a separate section, controlled by our Sector. Hence, the Sector agreed to forward a recommendation to the NTEP committee to grant the Sector a software-specific section of Publication 14. Accompanying this recommendation was an outline of the potential content that would be included. Full text of the recommendation is below:

Current state:

There is no single Publication 14 device category in which to place software-specific requirements, design considerations related to software or test procedures specific to software. Since most modern measurement devices contain software, to appropriately address any concerns each section of Publication 14 must include all software considerations. Further, each device section has a different governing Sector, which makes the process of change an exercise in convincing each Sector to make needed additions while keeping those additions harmonized across Sectors; an effort that has proven very difficult and time consuming.

Since the Sectors don't meet simultaneously, often our submissions are accepted into each Sector's agenda, then one will adopt and another will have comments or reject the request, leading to inconsistent treatment of software between classes of device.

Internationally, OIML and WELMEC have adopted a similar approach by segregating software recommendations/requirements into a standalone document or documents, and that approach aids both evaluators and submitters by consolidating the requirements for software into a single section that can be shared with developers.

Software Sector Proposal:

Create a Publication 14 Software category, which includes requirements, considerations and test procedures common to all software-based devices, including software-only products. Such a section might include the following:

1. *Models to be submitted for evaluation*
    a. *Determining scope of software to be approved*
        i. *Measurement and presentation*
        ii. *Calculations based on a measured value*
        iii. *Manual entry of measured value*
        iv. *Other*
    b. *Application of software may lead to additional Pub. 14 section consideration*
    c. *Minimum computing requirements statement*
2. *Software Identification*
    a. *Appropriate means of 'marking' metrologically significant software*
    b. *Software Separation and marking consequences*
    c. *Relationship between software and software identifier*
    d. *Presentation of software identifier*
        i. *Example icons and menu text*
        ii. *Exceptions*
3. *Protection against unauthorized software change*

   a. *How is software "sealed"?*
   b. *Remote software update considerations*
   c. *Audit trail (if employed) requirements for software updates*
4. *Accuracy of data calculations*
   a. *When to stop evaluating calculations & data manipulation*
5. *Software Evaluation Checklist*

Future Topics

1. *Distributed software considerations*
   a. *Securing communications between metrologically significant distributed software modules or components of a system*

It seems likely that action may take place within the next year, and that means the Sector faces the task of quickly publishing the text of a new section. It is hoped that some time could be spent developing the outline further and identifying content already created/included in other sectors that would need to be migrated to the new Section.

Discussion:
James Cassidy assured Jim Pettinato at the Annual Meeting this summer that they will take this under consideration. Darrell Flocken reported that the delay was due to not receiving input from the various sectors, either for or against. Darrell Flocken and Jim Truex are urging the various members to voice their opinion.

Some of the other sections of Pub. 14 already have software requirements, and there have been some questions regarding whether this would be removed and placed in the new software section. Jim Pettinato clarified that device-specific software requirements would remain where they are. The new software section would be more generic in nature.

SMA representatives indicated that their group may possibly review this proposal and come up with a position on the subject.

In the international community, there are general guidelines for software, such as in D-31, which are then adapted and implemented in the device-specific documents.

The starting point for the new software section in Pub. 14 would be the software checklist.

The new section would not be intended for software-only applications; it would be intended for anything metrological that has software.

There should be an introduction explaining when this section applies. “This code applies to the following… This code does not apply to the following…”

1. *Scope of application – any device of whatever type that contains software must meet the requirements herein. This includes both built-for-purpose and not-built-for-purpose software.*
2. *Materials to be submitted for evaluation*
   a. *Determining which software modules need to be approved*
      i. *Measurement and presentation*
      ii. *Calculations based on a measured value*
      iii. *Manual entry of measured value (e.g. measurement data rather than a measurement result)*
      iv. *Other*
   b. *Application of software may lead to additional Pub. 14 section consideration*
   c. *Minimum computing requirements statement*
3. *Software Identification*
   a. *Appropriate means of ‘marking’ metrologically significant software*
   b. *Software Separation and marking consequences*

   - c. *Relationship between software and software identifier*
   - d. *Presentation of software identifier*
      - i. *Example icons and menu text*
      - ii. *Exceptions*
4. *Protection against unauthorized software change*
   - a. *How is software "sealed"?*
   - b. *Remote software update considerations, e.g. authentication*
   - c. *Audit trail (if employed) requirements for software updates*
5. *Accuracy of data calculations*
   - a. *When to stop evaluating calculations & data manipulation*
6. *Software Evaluation Checklist*

Gathering some of the text we've proposed all in one place:

(3) The version or revision identifier shall be directly and inseparably linked to the software itself. The version or revision identifier may consist of more than one part, but at least one part shall be dedicated to the metrologically significant software.
[Nonretroactive as of January 1, 201X]
(Added 20XX)

Additionally, for software-based devices:

- **A description of the software functions that are metrologically significant, meaning of the data, etc., e.g. an architecture diagram or flowchart.**
- **The software identification (version, revision, etc.) , how to view it, and how it is tied to the software.**
- **An overview of the security aspects of the software(s), e.g. protection, user accounts, privileges, platforms, etc.**

G-S.9. Metrologically Significant Software Updates
A software update that changes the metrologically significant software shall be considered a sealable event.

It was suggested that we explicitly state that if something doesn't affect the metrological operation of a software-based device, we don't care about it.

It was suggested that we include a description of what information would be logged in a category 3 audit trail that pertains to software updates. What about category 2? Darrell Flocken recommended that we stay away from requiring any particular type of sealing category. For example, "When using a category 3 audit trail, the following information should be…" This would be a description of the methods to comply with the existing sealing requirements, not creating new requirements.

Mexico has a very thorough description of what is required in their audit trails. We may want to review that at some point.

We should incorporate the description of software separation from Doug Bliss' presentation.

Jim Pettinato suggested that we review some of the Software Sector meeting agendas from previous years for descriptions of exceptions and examples. Darrell Flocken will check to see if there is anything useful in the meeting agendas from the previous incarnation of the Software Sector. The D-31 document may be a good source of examples and explanations for issues to consider when performing a remote update.

Regarding the accuracy of calculations and at what point do you stop requiring evaluation, Darrell Flocken said that there's not a lot of existing documentation. The only guidance he thought HB44 includes on accuracy is regarding rounding. That's not the same thing as to when you stop the evaluation. "First final" is NTEP's standard, but the states can be different, requiring more. "First final" is in the Administrative Policy. The agreement as to where the boundary line is drawn may come about as a result of the discussion during type evaluation, but we can hopefully provide some guidance. This can be especially confusing when data is being transmitted and calculations are being performed remote to where a measurement was originally taken. HB44 deals particularly with “first final”, but how that interacts with HB130 (method of sale) can introduce complications.

Measurement Canada considers similar issues, requiring W&M regulation to the equivalent of our “first final”. Anything past that point isn’t metrological.

Conclusion:
The Sector concluded that we should organize and summarize the data captured in this brainstorming session on what will likely go into this new software section of Pub. 14. Teri Gulke volunteered to write a first draft for the Software Sector members to review and amend. Once the Sector has approved a draft of representative example content, we could choose to include this as an amendment to the NTEP agenda items.

8. Review/Discussion of new WELMEC 7.3/7.4 Drafts

Background:
WELMEC has been working on additional guidance for system architecture and design of software systems based on WELMEC 7.2 and has released two new draft guides titled ‘WELMEC Guide 7.3 Reference Architectures’ and WELMEC Guide 7.4 Exemplary Applications of WG 7.2’ for review by the wider group. These address some of the questions that have come up in our own discussions, such as cloud-based metrology, remote storage and displays, etc. Time permitting, the Sector can review this draft document and we can forward any additional comments to the Convener for consideration in their upcoming Group 7 meeting in Berlin.

Discussion:
There was some concern expressed that the text of the new draft guides may be too specific. For example, in WELMEC 7.3 there is a description of “pairing” a sensor with the software and how to accomplish it. It would be better to be more generic, and refer to “authentication” and “integrity” to establish a secure connection, rather than a particular method.

WELMEC 7.4 are oddball examples that may cause issues. Its title is “Exemplary Applications”. A better title may be “Anomalous Applications”. The second example is puzzling. They may be trying to describe a way to indicate that a measurement may not be accurate, but it’s not coming through clearly. There didn’t appear to be any authentication when a connection is established.

Discussion:
Our concerns will be relayed to the WELMEC working group via the CECOD representative.

9. Next Meeting

Background:
The sector is on a yearly schedule for NTEP Software Sector Meetings. Now that we've adopted a joint meeting system, the next Sector joint meeting will likely coincide with one of the remaining Sector meetings. The Measuring Sector would be next in the sequence if we continue in the same manner.

Discussion:
We are due to meet with the Measuring Sector next year. Their meeting will be next September in Denver. Between now and then, the conference will meet twice, so the addition of a new software section within Pub. 14 may have been addressed by that time.

We discussed whether it was still beneficial to conduct joint meetings with the other sectors. Doug Bliss is retiring, so this is his last Software Sector meeting. Darrell Flocken asked whether we intend to replace him. Jim Pettinato asked about the standard of having a NIST/NTEP technical advisor. Darrell Flocken said that there is discussion of moving away from that standard and adopting Software Sector's example of having technical advisors from industry.

The next meeting should have an agenda item for appointing a new technical advisor. If we could do that prior to the next meeting, that would be even better. Perhaps the nominations could be conducted via email.

Conclusion:
We agreed to continue with joint meetings for at least one more year (2019). After that meeting, we may want to consider conducting joint meetings with the NTEP labs.

Appendix A – Acceptable Menu Text/Icons for Weights Measures information

Permitted Menu Text examples	*Permitted Icon shape examples*	*Essential characteristics*
Information Info	i i i	Top level menu text or icon • Icon text is a lower case "i" with block serifs • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Help ?	? ?	Top level menu text or icon • Icon text is a question mark • Text color may be light or dark but must contrast with the background color • Icon may have a circular border • Activation of this menu text/icon may invoke a second level menu text/icon that recalls metrology information.
Metrology Metrological Information	M M	Top or second level menu text or icon • Icon text is an upper case "M" • Text color may be light or dark but must contrast with the background color • Icon may have a circular, rectangular, or rounded rectangle border. • If present, the activation of this menu text/icon must recall at a minimum the NTEP CC number.
NTEP Data N.T.E.P. Certificate	NATIONAL CONFERENCE ON WEIGHTS AND MEASURES NTEP	This one is debatable – what if the certificate is revoked? Does NTEP grant holders of CCs the right to display the logo on the device, or just in documentation?
Weights & Measures Info	W&M W/M	

Appendix B – NIST WMD Report on International Activity

Summary of OIML D31 Revision Progress
To be presented at NCWM Software Sector Meeting, Louisville, KY
K.M. Delak 17 Aug 2018

OIML has been undertaking a revision of D-31: General Requirements for Software Controlled Measuring Instruments. This falls under Technical Committee 5, Subcommittee 2. Approval of the revision was taken at the CIML meeting in October 2016, and initial work began in spring of 2017, with the 1WD being circulated at that time for input.

September 2017: Project group met in Berlin to consolidate comments and complete a first revision. This constituted 1CD, which was subsequently circulated for vote and comment. *Circulation of a CD was chosen to ensure a maximum response from OIML member countries.* Further, two subgroups were formed: (1) Software Verification and (2) Operating Systems.

Discussions on draft language for Operating Systems were conducted largely between only the US and Germany by video-conference over the course of two months. The consensus language generated from this activity was introduced into the document in the subsequent project meeting.

Draft language for Software Verification was agreed to primarily by correspondence. This also was introduced into the document draft in the subsequent project meeting.

May 2018: Project group met in Dordrecht to consolidate comments from the 1CD. The group made rapid progress in consolidating language. The conveners initiated a third subgroup, Terminology Harmonization, to clarify the definitions on “measurement,” “measurement result” and “measurement data.” Current suggestions have been circulated to the USNWG for comment. The conveners ask for finalization of input to this by 24 Aug 2018.

It is expected that 2CD will be published in September. This will also be circulated to the USNWG for final comment and vote.

WELMEC WG7 has attempted to further clarify interpretation of WELMEC 7.2 with new draft documents WELMEC Guide 7.3 "Reference Architectures" and WELMEC Guide 7.4 "Exemplary Applications", meeting coming up on 8/29/18 @ PTB offices in Berlin.

US National Working Group consists of:

Dr. Katya Delak
Jim Pettinato
Doug Bliss
Teri Gulke
Jan Konijnenburg
Joe Porthouse
Shakila Xavier