National Type Evaluation Program (NTEP)
Software Sector Meeting Summary

May 6th, 2021 / Online
In conjunction with the Multiple Dimensioning Measuring Device (MDMD) work group meeting

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 **<u>underlining</u>** information to be added. Requirements that are proposed to be non-retroactive are printed in ***bold faced italics***.

Table A
Table of Contents

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 is a joint meeting with the MDMD work group, some time will be allocated to meet and greet both new and familiar faces.

STATUS REPORTS – RELATED NCWM AND INTERNATIONAL ACTIVITY

Attendees of the 2021 NCWM Interim and Annual Meeting are 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.

At the 2021 Interim NTEP Committee Meeting, the committee reiterated that they're looking forward to seeing our first draft of Pub. 14.

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 groups. 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 importance or interest should be allocated more time during the joint session day.

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.

John Roach asked whether CC's will still allow "and higher" terminology in reference to the version numbers. International standards often do not allow that, though bug fixes can be an exception. The risk class affects how much they scrutinize changes. Darrell Flocken said that we should be creative when handling software changes, so that we aren't requiring unnecessary reevaluations. Until an alternative means of managing firmware updates (both significant and non-significant metrologically) is codified, it is likely unavoidable to continue allowing 'and higher'.

Tina Butcher pointed out that the requirements regarding audit trails ensure that the audit trail is isolated to only include metrologically significant information. This may be applicable to viewing the version and CC. The access to view the information must be separate from the ability to change the metrologically significant parameters. There's a lot more information in Pub. 14 than in HB44, though field inspectors aren't likely to rely heavily on Pub. 14.

The Software Sector's Pub. 14 document should include guidance on audit trails – accessing, viewing, etc. NIST has a training video on this subject. Pub. 14 for the Measuring Sector Appendix A&B and Weighing Sector's Appendix B currently include guidance on audit trails.

It was suggested we include an indication of preference for how to mark the CC. That is, indicate that the preference is for continuous display, failing that, making it easy to access from a limited number of options. Hard marking is least preferred. In other words, document best practices.

Jan Konijnenburg asked how an inspector can get to the certificate number if it isn't hard-marked or continuously displayed. If there are a limited number of options leading to displaying it, that helps, though it could be challenging to reach a consensus on the allowed options.

It was noted that most inspectors don't like to access multiple screens. They would much prefer to just have the CC number continuously displayed.

Software that runs on tablets tends to have extremely limited real estate on their screens. It's very difficult to continuously display a CC number on that sort of app. A limited set of text or icons should be something most inspectors can adapt to. There's a concern that the inspector might accidentally change a setting; this also applies to inspectors looking for the audit trail or software version number.

If we can alter G-S.1.e. sufficiently, we may be able to eliminate G-S.1.1.

Darrell Flocken recommended that we begin working on this item prior to 2022 given that it may take some time for others to accept any changes we propose. We will create subgroup to work on a proposal including Jim Pettinato, Vere Miller, Teri Gulke, Darrell Flocken, and Tina Butcher (as a consultant/reviewer). Jim Pettinato will send out an invitation and set up a meeting.

Discussion:
G-S.1.1. Location of Marking Information continues to use the terminology "Not-Built-For-Purpose". We would prefer to reduce the usage of that term and "Built-For-Purpose" (or eliminate them altogether). Those categories continue to blur as time goes by. It was acknowledged that it is always more difficult to alter the general code.

Darrell Flocken reported that NTEP is challenged when they get software running on e.g. a phone or tablet. Developers have sometimes failed to properly display the version number, which makes it difficult for inspectors to view the information. Pub. 14 should indicate that continuously displaying the version number is the most preferred method. HB44 has the requirement, and Pub. 14 is for explaining how to comply.

The group agreed with the interpretation that 'Continuously displayed' is intended to mean while in operating mode. The CC has to be permanently marked or continuously displayed. This allows the information for accessing the version number to be within the CC.

Zach Tripoulas asked whether the group thought that the Scale Marking Requirements are congruent with G-S.1. It was pointed out that some of the exceptions noted in G-S.1. are intended to cover applications that can't comply with the general requirements. One example is 7-segment displays. Refer to G-S.1.d.1.i. and G-S.1.d.1.ii.

Darrell Flocken recommends that we plan a review of the contents of HB44 to verify that it correctly reflects our understanding of the intent. From there, we can clarify matters within Pub. 14, potentially within a checklist. Then we can come back to determine whether any changes are needed to HB44.

Conclusion:
Jim Pettinato expressed concern that eliminating differences between Not-Built-For-Purpose and Built-For-Purpose will require rewriting rather than minor tweaks to HB44. The Sector agreed to focus on Pub. 14 for now.

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, 202X]
(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, 202X]
(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".

This agenda item also was tabled until a decision on the direction for Publication 14 was made by the NTEP committee. Since we have been given the go-ahead to develop a section for Publication 14 specific to software, the Sector should finish developing this item.

We discussed how bug fixes will be handled when software version identifiers become inseparable from the software itself, such as when a checksum is used. Typically, the software change would need to be reported, and NCWM would decide whether the software would need to be reevaluated or a description simply added to the certificate.

A lot of this will be dependent on how the software identifier is defined. For example, for V1.xx.yy, yy could be used solely for bug fixes.

John Roach reports that his lab only allows the methods described in G-S.1.1.b or on the list in Appendix A. Teri Gulke reports that LC's Marketing person dislikes the example icons and finds them limiting.

Jim Pettinato asked what the opinion of the labs and field inspectors is regarding whether the requirement to make the software identifier inseparably linked to the software must be included in HB44. Darrell Flocken didn't think it was intuitive to look under 'G-S.1.1. Markings' for this requirement. It probably should be in the general code of HB44, but perhaps somewhere other than Markings. Tina Butcher said that G-S.1.1. is a bit of a mess, and would

benefit from being reorganized. Assuming we do eventually get rid of the differentiation between built-for-purpose and not-built-for-purpose, that will certainly help in streamlining (or eliminating) G-S.1.1.

The Sector will prioritize work on the Pub. 14 software section. In 2021, we will consider revisions to G-S.1.1 as well as the changes pending as described in Agenda Item 1, since the non-retroactive dates will be expiring.

Also in 2021, the example menu text/icons in Appendix A need to be revisited to update/extend the list and clarify the application (just examples, or complete set of allowed solutions?)

Now that the Software Sector has its own Pub. 14, the question was raised as to whether the proposed text need to be part of Handbook 44. Darrell Flocken recommended that the Sector continue to move forward this item with the goal of inclusion of the proposed text into HB44. There was general consensus on this approach.

The Sector will prioritize work on the Pub. 14 software section. We will consider revisions to G-S.1.1 as well as the changes pending as described in Agenda Item 1, since the non-retroactive dates will be expiring.

Discussion:
Jim Pettinato and Darrell Flocken both expressed the opinion that these recommendations and information regarding software separation do not constitute a marking requirement. G-S.1. already includes a marking requirement for a version / revision.

This wording has already been incorporated into Pub. 14's for the various sectors, prior to the creation of a Software Pub. 14. Eventually Darrell Flocken will have to go back and convince the other sectors to remove the redundant wordage from their Pub. 14s.

Conclusion:
Once the Sector has satisfactorily included language within Pub. 14 to address this point, we will consider this agenda item finalized.

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. <u>This section and section 4 are intended to be mutually exclusive. Complete this section only if you replied Yes to 1.1.</u>

3.3. 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. <u>Complete this section only if you replied No to 1.1.</u>

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). **<u>This is a declaration or explanation by the manufacturer.</u>** ☐ 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). **<u>This is a declaration or explanation by the manufacturer.</u>** ☐ 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. **<u>If software separation is employed, t</u>**he program modules of the metrologically significant software are defined and separated. ☐ Yes ☐ No ☐ N/A

5.10.2. **<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

5.10.3. 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

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 an implementation meets type.

Darrell Flocken offered to start formalizing the procedure for distributing the checklist to submitters. The Sector will work on crafting an explanation for the NTEP labs as to how the answers to the checklist benefit their inspectors.

At the 2019 meeting Darrell Flocken provided 5 returned checklists, as well as some feedback on the checklist. Somewhat surprisingly, of those submitting completed checklists, 4 of the 5 reported that they're performing software separation. The feedback was largely from scale manufacturers, so there wasn't much feedback related to software/firmware updating or revision tracking. There appears to be a lot more software-only (e.g. 'app') device type applications for certification. They typically perform software separation and are downloadable.

Some specific feedback was given by one submitter on the checklist itself. Points of comment included the following:

- Checklist items 1.1. and 1.2 independently might make sense, but together seem contradictory.
- The term "user" might be inaccurate in some applications, suggest using "consumer"
- Likewise, "manufacturer" might a better term to use than "submitter"

Additional feedback from the labs was that it still seems that software developers often didn't understand the questions on the checklist. This was particularly true for app developers who don't have any experience with W&M.

There was a bit of confusion regarding the version of the checklist that people have copies of. Jim Pettinato suggested adding a version number to the checklist.

The revised checklist was included in the draft Pub 14 section and sent to the Sector for review.

In 2020 Darrell Flocken reported that he's only received one checklist this year. It was a software-based company, and they seemed to understand the points on the checklist. He suggested that we change the checklist from

something that we've been using to gauge understanding in industry to something that is primarily intended to provide guidance to companies that have limited W&M knowledge.

Jim Pettinato suggested we explore adapting text from international documents to provide such guidance. We need to evaluate the effort required, and how much value would be provided. Darrell Flocken suggested that a white paper could be disseminated via the NTEP newsletter. It could also be linked to from the NTEP website.

John Roach reported that he has a company that's intending to upgrade software, and they say that their metrologically significant software is separated from the non-metrological software. He's looking for a bit of guidance on how to handle the situation. Jan Konijnenburg pointed out that it's difficult to verify they're doing things correctly just looking at the software itself. Documentation plays a large role in showing inspectors / evaluators that the design has addressed the software separation properly.

It was agreed that additional guidance would be beneficial for the application of the checklist. Either a guidance document or a white paper or both to accompany the checklist will be assigned as work items for a future Software Sector meeting.

Discussion:
It seems like the feedback from Sector members indicates that it would be beneficial to the checklist to add some explanatory text. Discussion on how to best implement this explanatory text included suggestions to incorporate information from international standards as one possibility.

Darrell Flocken received only one checklist since our last meeting. At the March lab meeting he asked whether people were still handing them out; they are, but they're not getting anything back. NTEP needs a software application that includes the policy, but we need to complete the policy first.

Conclusion:
The Sector will table this agenda item until the next meeting. Completing the Software Policy document will be the priority.

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.

Existing documentation required for obtaining certification is outlined in Pub. 14, administrative policy 9.1.7:

- 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.

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.

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.

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.

The Software Sector recommended that the following 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 and data, etc. that are metrologically significant, 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.**

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.
Since we will be drafting a new section for Pub. 14 on software, we discussed where it should go within Pub. 14. It was suggested it may best fit as a standalone section following the Administrative Policy. It was noted that each specific device section for Pub. 14 might need a statement saying that manufacturers should reference the software section of Pub. 14. Darrell Flocken didn't think the software section will be big enough initially to support its own separate book. Ideally, the software section wouldn't cost extra.

Jeff Gibson was concerned about what level of detail some of this documentation would reach since, if it's too detailed, it might not be helpful for the NTEP evaluator. It might be too difficult to understand. Darrell Flocken mentioned that last year the thought had been that the Software Sector might give a presentation at the Lab meeting; however, that didn't work out. He thought that it might be better to wait a bit longer. Perhaps after the draft for Pub. 14 is written and reviewed, that might be a better time for a presentation to the Labs.

The Sector recommended a standalone section, following the Administrative Policy, if possible. The above list of suggested documentation would be included within the new software section of Pub. 14.

Darrell Flocken commented that NTEP has been receiving a lot of software-only applications. Note that we have included the language shown above in Section 3 of our draft Pub. 14 section already.

Discussion:
In practice, there might not be just one application. Instead there might be device-specific applications, e.g. a scale software application. Instead of multiple new applications to address these related packages, perhaps it could be handled as a sub-section added to an existing application. There's a meeting coming up to address the direction of applications which will probably affect matters.

Conclusion:
We will await the outcome of the upcoming NTEP meeting and progress accordingly. The assumption is that the proposed language will be part of the Software Policy document.

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. 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.

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.

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?

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.

It was noted that recommendations for changes to HB112 should go to Tina Butcher.

Allen Katalinic asked how many field inspectors typically checked more than the certificates on the pump, its calibration, and the receipts. It sounds like they usually don't check the POS system. They generally rely upon the CC as an indication that the system has passed type approval and can thus be assumed to work properly. CA is an exception. It sounds like Maryland has been diligent in checking version numbers and has, on at least one occasion, found a problem.

It is difficult to obtain feedback from the Regional meetings since representatives of the Software Sector don't necessarily attend those meetings. Another suggestion would be to include information in the newsletter. This actually came up as a suggestion in 2018 as well.

There had been a request for training on the NTEP process. Darrell Flocken has provided a presentation on this subject in the past, but it may be time for a refresher.

NIST does perform regional training for field inspectors. Sometimes they'll bring a subject matter expert along to assist with the training. Tina Butcher listed several training courses that have been given recently.

Tina Butcher said that there have been requests for training schools in conjunction with the regional meetings. They're sometimes also looking for presentations. Unfortunately, the majority of the attendees aren't necessarily field inspectors.

John Roach said that if we trained CA's main trainers, that information could trickle down to the field inspectors. Jeff Gibson said that a similar approach would work with OH.

It sounds like we'd really benefit from someone making a presentation that could be disseminated. Jan Konijnenburg and Tara Pandey volunteered to draft a presentation. Jim Pettinato will meet with them to discuss what will need to be included. If we have time on Thursday after working on the Pub. 14 draft, we could consider working on this presentation.

Tina Butcher suggested recording a webinar.

A presentation on software versioning targeted field inspectors will be developed. Time permitting, we will use part of the 2nd day as a working session to further the development of both the draft Pub 14 Software document and an initial training presentation. If possible, when this presentation is given we could record it and make it available for NIST or NCWM to use as online training material.

Jim Pettinato mentioned that we do have a draft training presentation. It needs a bit of cleanup, and he hasn't had a chance yet to share it with Tina Butcher. Jim volunteered to update the draft template and add some additional content (e.g. Example 3 in the presentation). He also mentioned that if any manufacturers want to submit additional examples, they could be included.

Discussion:
Jim Pettinato updated the training presentation. Additional examples would be welcome. Chris Senneff volunteered to send Darrell Flocken / Jim Pettinato his certificate.

Darrell Flocken recommends that we participate in the Lab Meeting and discuss this information with them.

Conclusion:
Any additional information received regarding specific examples of the implementation of features to support field inspection in real-world devices will be incorporated into the work-in-progress presentation. If possible, we will arrange to have representation at the NTEP lab meeting as well.

6. 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.

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.

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.

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.

We allocated a portion of the 2nd day to a working session on the Pub. 14 draft section. The draft was circulated for comment, and several participants provided feedback which has been/will be incorporated.

As discussed earlier, Section 3 could benefit from being fleshed out further.

John Roach said that they are receiving many software-based applications; however, they often have little to do with metrology – e.g. operational functions, agreement of indication.

Darrell Flocken pointed out that other Sectors' Pub. 14 often start in manner similar to ours; however, they often end their various sections with a checkbox intended to show that the sections were acknowledged and/or applicable. Jim Pettinato agreed that this sort of format could be useful, and it would be a good idea to match the other Pub. 14's formatting. Alternatively, checkboxes of that sort could be isolated in an appendix.

We further discussed the path forward for the software-specific section of Pub. 14. Darrell Flocken asked for opinions on whether we want it to be a stand-alone document, or included at the beginning of each document, or as part of (following) the Administrative Policy portion? Darrell Flocken recommended that we promote the Software Policy section as our own separate document; otherwise, it could prove complicated to make updates to our Pub. 14.

Also, it was noted that the Administrative Policy is free, but the other documents have fees. Jim Pettinato recommended that we not charge for our document since it would be disseminated in conjunction with for-a-fee device-specific sections.

Consensus of the Sector is that we will have a separate document for our Pub. 14, and it will be free. We will leave it up to Darrell Flocken as to how to best distribute it – probably a free download from the website.

Darrell Flocken said that he'll review our Pub. 14 in detail over the next month to identify elements that do not currently have support in HB44. Subsequent to that, Jim Pettinato can work on a draft to address those concerns. Jan Konijnenburg suggested that the Pub. 14 Sections have references to HB44.

Discussion:
It was pointed out that the title of the document should not include the word 'Administrative'. Instead we will use the title 'Software Technical Policy'. The draft document title and header needs to change to Software Technical Policy. We'll edit it offline since the document has some editing issues.

Section 3 requires training, so that the labs all have a common understanding of what is needs. Darrell Flocken isn't certain that the third bullet point may imply that the NTEP evaluators will be reading a flowchart in detail. He thinks the labs might not be ready for that, and the evaluation process doesn't go that deep for now. That doesn't mean that we should remove the bullet point since it's beneficial to have the submitters ensure that they've thought about the issues.

Patrick Tilley asked about remote software upgrades. Darrell Flocken replied that the software upgrade would need to be recorded in the audit trail as an event log entry. Jan Konijnenburg explained the OIML approach used in Europe and how different countries can handle it differently. Ron Peasley also pointed out that it's device-specific in Europe, dependent on risk category.

Teri Gulke will ask her company's marketing if they can provide additional examples of potential icons to use to access the version number.

Conclusion:
Jim Pettinato will make the identified edits to the draft Software Technical Policy and circulate to the Sector for comments.

7. 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.

If we continue with our joint meetings, 2022 would be in conjunction with the Weighing Sector.

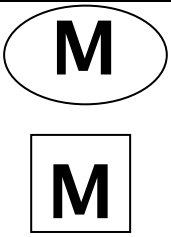
Discussion:
The date for the 2022 Weighing Sector meeting hasn't been picked yet, but it's usually the second or third week of August. The location of the meeting has yet to be determined. Also, the Belt Conveyor Sector's meeting will be merged with the Weighing Sector's meeting, causing it to probably extend an extra 4 hours.

We discussed whether it continues to make sense to conduct joint meetings. Teri Gulke and Sprague Ackley spoke in favor of continuing with joint meetings. The were no objections, so we will request a joint meeting in 2022.

Jan Konijnenburg suggested that we actually meet with the Weighing Sector in 2021 in San Antonio, TX (the location is still tentative), to speed up the process. Darrell Flocken said that this wouldn't add too much cost, and he'd have to get approval from the NTEP Committee. Alternatively, we could setup an additional Zoom meeting, or set up a working group. Other time commitments could be a problem.

Conclusion:
We'll follow up after Darrell Flocken talks to the NTEP Committee, but the Sector requested to hold the next meeting in conjunction with the Weighing Sector in August 2022.

Appendix A – Acceptable Menu Text/Icons for Weights Measures information

Permitted Menu Text examples	*Permitted Icon shape examples*	*Essential characteristics*
Information Info		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		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		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

Dr. Katya M. Delak

OIML Document D31: General Requirements for Software Controlled Measuring Instruments was approved for publication in October 2019 at the CIML meeting. It subsequently went into immediate revision to address new and evolving areas related to software-controlled instruments that were not addressed in the previous document. This revision is being carried out under OIML TC 5/SC 2/p 4, whose project lead is Marko Esche of PTB (Germany).

In the 2020-2021 timeframe, three separate subgroups met periodically to work on annexes and textual revisions related to artificial intelligence and machine learning (Subgroup 1), remote verification (Subgroup 2) and terminology (Subgroup 3). Proposed texts and changes were shared with the USNWG, but ultimately, most of the derived language was developed by other participating members of the subgroups: Germany, Switzerland, Australia, Czech Republic, Netherlands, Korea, Japan, and Slovenia.

At the same time, a first working draft (1WD) was circulated to project group members for comments, which closed on February 15th. This working draft provided an opportunity for primarily housekeeping revisions and editing of the existing D31 document.

The project group will meet this month next week, May 10-12th from 7:00 AM – 9:00 AM EDT. The meeting will cover the adjudication of comments to 1WD and will also provide an opportunity to the three subgroups to present their work.

US National Working Group consists of:
Katya Delak
Jim Pettinato
Teri Gulke
Jan Konijnenburg
Joe Porthouse
Todd Gray