Software Validation in Metrology
(Experience from PTB’s Activities)

Dieter Richter
(Norbert Greif, Ulrich Grottker)
Physikalisch-Technische Bundesanstalt Braunschweig and Berlin (PTB)
Berlin, Germany

(Dieter.Richter@ptb.de)
When thinking of the impact of IT in metrology, one has at first enhanced (metrological) methods, enhanced standards, etc., in mind.

However, there is another impact:

Metrology is faced with new types of (mainly) horizontal problems as, e.g.,
- software validation,
- (distributed) databases,
- security of software and data,
- …
Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Outline

Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Software testing and quality assurance

Accredited Testing Laboratory

in accordance with ISO/IEC 17025

Issue of independent software test reports

Two units under one roof

IT in Legal Metrology

Support of type approval testing, provision of guidelines and methods
Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Why is software validation so important?

- Fit-for-purpose software
- Reliability of software
- Reuse of software
- ....
Software Validation: Initial Question 2

Why is a common understanding of software validation so important?
Why is a common understanding of software validation so important?

Imagine following situation:
A software test has been performed (e.g. a functional test).
Results are available.
What does the test results tell?
Why is a common understanding of software validation so important?

Imagine following situation:
A software test has been performed (e.g. a functional test).
Results are available.
What does the test results tell?

- Whom shall the results anything tell? (The tester? Anybody else? How to communicate?)
- Does the results tell that the software is fit for the intended purpose? (What is exactly the purpose?)
- Is the software reusable (For what tasks?)
- Are the test results trustworthy? (Was the appropriate test method used? Was the test amount appropriate?)
Do we have a common understanding of software validation?

Obviously not!

This causes difficulties for
- performing validations,
- reporting on validation,
- …

→ A common understanding is necessary.
Software Validation: Efforts Undertaken

Systematic Approaches

- **PTB**: Increasing number of particular working instructions
- **NPL**: Best practice guide: Validation of software in measuring systems
- **Europe**: Software guide for legal metrology (WELMEC Guide 7.2)
- **USA**: FDA recommendations
- ...
What is the basis for a common understanding?
Definition (ISO 9000:2000):

“Confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled”.

- (1) Requirements must be known.
- (2) Objective evidence must be provided.
Software Validation: The Procedure

1. Definition of requirements
   (e.g. numerical stability)
   + Refinement

2. Selection of methods
   (e.g. dynamic test with appropriate test data)

3. Carrying out methods
   ....
   ....
   ....

4. Documentation of results
   (e.g. test report including test data sets)
Outline

Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Example of Requirement Refinement: Gauge Block Calibration (1)

Software
requirement
Class

Functionality of control software

... ... ...

Metrological Specification

Plausibility of automatic data reading
Plausibility of manual data input
Correctness of data processing
Conformity to ISO 3650

...
Example of Requirement Refinement: Gauge Block Calibration (2)

Metrological Specification

Further refinement steps for achieving testablity

Correctness of data processing

Reproducibility of known length differences

Correct treatment of parameters

No storage of data if measurements is incomplete

No increase of uncertainty by software

Observation of validity of calibration data

Correct processing of temperature data

Correct processing of user-specific tolerances

...
### Example of Requirement Refinement: Gauge Block Calibration (3)

<table>
<thead>
<tr>
<th>Test procedures</th>
<th>Test methods</th>
<th>Further refinement steps for achieving testability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use of program with test data</td>
<td>Observation of validity of calibration data</td>
<td>...</td>
</tr>
<tr>
<td>Black-box test</td>
<td>Correct processing of temperature data</td>
<td>...</td>
</tr>
<tr>
<td>Code inspection</td>
<td>Correct processing of user-specific tolerances</td>
<td>...</td>
</tr>
<tr>
<td>Code inspection</td>
<td>Inspection of the respective part of code</td>
<td>...</td>
</tr>
<tr>
<td>Code inspection</td>
<td>Inspection of the respective part of the code</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Correct treatment of parameters**

---

NMIJ-BIPM Workshop on the Impact of Information Technology, Tsukuba, Japan, 18-20 May 2005  
Richter 3-3
Are there typical requirements for metrological software?
Metrological Software: Omnipresence

- Data acquisition
- Calibration
- Verification
- Data transmission
- Simulation of measurements
- Testing
- Analysis of data
- Control of instruments
- Maintenance

NMIJ-BIPM Workshop on the Impact of Information Technology, Tsukuba, Japan, 18-20 May 2005
<table>
<thead>
<tr>
<th>Type of requirements</th>
<th>Type of validation methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Model conformity</td>
<td>- Comparison of experiment and simulation</td>
</tr>
<tr>
<td>- Standard conformity</td>
<td>- System test, black-box test</td>
</tr>
<tr>
<td>- correctness of implementation</td>
<td>- White-box test, code inspection</td>
</tr>
<tr>
<td>- numerical stability</td>
<td>- Expert evaluation</td>
</tr>
<tr>
<td>- software performance</td>
<td>- Field experience, long-term experience</td>
</tr>
<tr>
<td>- software and data security</td>
<td>- Acceptance of QM system</td>
</tr>
<tr>
<td>- usability of software</td>
<td>- Design review</td>
</tr>
<tr>
<td>- maintainability of software</td>
<td>- Performance measurement</td>
</tr>
</tbody>
</table>
Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Software for

- Calibration of roughness standards
- Control of weighing machines
- Calculation of measurement uncertainty
- Control of speed measurement and transmission of data via networks
- Calculation of radiation dose
## Software Validation: Exemplary Examples carried out at PTB

<table>
<thead>
<tr>
<th>Subject of software</th>
<th>Key requirements</th>
<th>Customer / Cause of Validation</th>
<th>Methods used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration of roughness standards</td>
<td>Conformity to field-specific standards</td>
<td>PTB lab / Requirement of lab’s customers</td>
<td>Dynamic tests with test cases, doc inspection, code inspection (partially)</td>
</tr>
<tr>
<td>Control of weighing machines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculation of measurement uncertainty</td>
<td>Conformity to GUM</td>
<td>PTB and calibration labs / ISO 17025 (Acceptance test before purchasing)</td>
<td>Dynamic tests with test cases, doc inspection, code inspection (partially)</td>
</tr>
<tr>
<td>Speed measurement and transmission of data via networks</td>
<td>(1) Correctness of measurement (2) Security of data</td>
<td>Company / Legal metrology regulation</td>
<td>(1) Functional tests (2) Code inspection both: doc inspection</td>
</tr>
<tr>
<td>Calculation of radiation dose</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Software Validation: Exemplary Examples carried out at PTB

<table>
<thead>
<tr>
<th>Subject of software</th>
<th>Key requirements</th>
<th>Customer / Cause of Validation</th>
<th>Methods used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration of roughness standards</td>
<td>Conformity to field-specific standards</td>
<td>PTB lab / Requirement of lab’s customers</td>
<td>Dynamic tests with test cases, doc inspection, code inspection (partially)</td>
</tr>
<tr>
<td>Control of weighing machines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculation of measurement uncertainty</td>
<td>Conformity to GUM</td>
<td>PTB and calibration labs / ISO 17025 (Acceptance test before purchasing)</td>
<td>Dynamic tests with test cases, doc inspection, code inspection (partially)</td>
</tr>
<tr>
<td>Speed measurement and transmission of data via networks</td>
<td>(1) Correctness of measurement (2) Security of data</td>
<td>Company / Legal metrology regulation</td>
<td>(1) Functional tests (2) Code inspection both: doc inspection</td>
</tr>
<tr>
<td>Calculation of radiation dose</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Software Validation: Exemplary Examples carried out at PTB

<table>
<thead>
<tr>
<th>Subject of software</th>
<th>Key requirements</th>
<th>Customer / Cause of Validation</th>
<th>Methods used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calibration of roughness standards</td>
<td>Conformity to field-specific standards</td>
<td>PTB lab / Requirement of lab’s customers</td>
<td>Dynamic tests with test cases, doc inspection, code inspection (partially)</td>
</tr>
<tr>
<td>Control of weighing machines</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculation of measurement uncertainty</td>
<td>Conformity to GUM</td>
<td>PTB and calibration labs / ISO 17025 (Acceptance test before purchasing)</td>
<td>Dynamic tests with test cases, doc inspection, code inspection (partially)</td>
</tr>
<tr>
<td>Speed measurement and transmission of data via networks</td>
<td>(1) Correctness of measurement</td>
<td>Company / Legal metrology regulation</td>
<td>(1) Functional tests (2) Code inspection both: doc inspection</td>
</tr>
<tr>
<td></td>
<td>(2) Security of data</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Calculation of radiation dose</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Motivation for validation

**Often:** Validation is over-directed or prescribed

**It happens:** Validation is a voluntarily requested (e.g. safeguarding of results)
Time and costs

Uniform desire: As less as possible

(Contradiction to technical assessment)
Observations during test execution

Often: Deviations from requirement specifications, software defects, inconsistencies

(⇒ Validation was necessary, problems of time and costs, …)
Quality of Requirements

**Sometimes:** specifically available, e.g. by reference (standard, guideline, ..), in particular in legal metrology

**Often:** vague as, e.g., “good quality of software”, “fit for purpose”, “correct”, “secure”

⇒ Begin of “negotiations” to refine the requirements
Test plan (selection of methods)

**In legal metrology:** With an increasing tendency, available in guiding documents derived from regulations

**In non-legal metrology:** Usually not available, needs to be derived from finally specified requirements
Outline

Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Basic Requirements for Universal Computers (PC):

Guide

Long-term storages

Extension L

Fuel Dispensers

Weighing Machines

Data Transmission

Extension T

Taxameters

Electricity Meters

Basic Requirements for Built-for-purpose Devices:

Guide

Shells of Requirements

Risk Classes

NMIJ-BIPM Workshop on the Impact of Information Technology, Tsukuba, Japan, 18-20 May 2005

Richter 5-1
MID-Software: P Requirements

P1 - Documentation
P2 - Software identification
P3 - Influence via user interfaces
P4 - Influence via communication interface
P5 - Protection against accidental or unintentional changes
P6 - Program protection against intentional changes
P7 - Parameter protection
**Recommendation for Validation**

**Risk class A:**
- none

**Risk class B and C:**
- check of documentation for admissibility of all user commands / user menu items
- require declaration of completeness
- carry out all documented commands

**Risk class D:**
in addition to C and D:
- check test protocols for appropriateness of test methods carried out by developers

**Risk class E and F:**
in addition to B and C:
- check software design for unambiguous definition of user commands
- check dataflow from input to internal function / protected area
MID-Software: Further Result

(see www.mid-software.org)
Challenge (1)

More complex, software-controlled instruments, remote activities, data distribution

Software-controlled manufacturing of measuring devices

Multidimensional and Weighing Measuring Instrument for Calculating Freight Rates

© Cargoscan
Challenge (2)

More complex, software-controlled instruments, remote activities, data distribution

Software-controlled manufacturing of measuring devices

Assembling / Configuration  Calibration  Packaging

Automatic Calibration of pressure sensors as part of manufacturing facilities
Developments in IT from Which Metrology Can Benefit

- The world of software standards
- Test engines and test standards
Developments in IT from Which Metrology Can Benefit

The world of software standards

Compatibility of software quality to other technical areas
The World of Software Standards

Software quality standards

Vocabulary standards

Product standards

Process standards

Laboratory related standards

ISO/IEC 17025
EA Guideline
OECD/GLP

Quality management
ISO 9001
ISO 9000-3

Software life cycle
ISO/IEC 12207
V-Model

Process assessment
ISO/IEC 15504
(SPICE), CMM

Programs and data
ISO/IEC 9126
ISO/IEC 12119

Documentation
ISO 6592
DIN 66230

Security safety
ISO/IEC 15408
CC, ITSEC

Usability
IEC 61508
ISO 9241

Developments in IT from Which Metrology Can Benefit

Test engines and test standards

Reduction of test costs and unambiguous communication on
/ documentation of validation
Test Standard (TTCN-3)

What is TTCN-3?

• **Standard** for the specification and implementation of tests (ETSI, ITU, under discussion in IEC)

• Appropriate for all types of **dynamic tests** (functional test)

• Applicable to independent software, systems with embedded software, distributed systems, communication systems (**reactive systems**)

• **description format** (language, tables, graphics) for
  - test data generation
  - test execution
  - **interfaces** / external functions
  - test report generation

→ Applicable to a great deal of metrological software
What is the benefit of TTCN-3 for metrological software?

- The standard is adaptable / can be tailored to the world of metrology.

- Terminology and particularities of metrology can be considered.

- A standard can significantly enhance the documentation of software validations.

- Ones one has an adapted test engine based on a standard, it saves time and costs for preparation, execution and reporting on software tests.
Validation of Metrological Software with TTCN-3 and test engine

**Expected support by**

- **Descriptive means**
- **Execution means**

**Decisions that can not / hardly be automated**

**Given:**
Software/system under test

**Definition of requirements and their refinements**

**Selection of test methods**

**Execution of tests**

**Provision of test data**

**System for the execution of tests**

**Evaluation of test results**

**Generation of test reports**

**Validation report**

NMIJ-BIPM Workshop on the Impact of Information Technology, Tsukuba, Japan, 18-20 May 2005  Richter  6-10
Outline

Software Validation in Metrology
(Experience from PTB’s Activities)

1. Background at PTB
2. Basic considerations on software validation
3. Requirements and validation methods
4. Examples and experience
5. Requirements in legal metrology
6. Future aspects
7. Summary
Summary: Software Validation in Metrology

• Mastery of the entire process:
  Determination of key requirements
  → Refinement to testable/observable attributes
  → Selection of appropriate test methods
  → Execution of tests
  → Documentation of Results
  → Issue of objective validation reports

• Validation is an interdisciplinary work
  - Metrologists and software engineers can well share the work.
  - They must cooperate at decisive points (e.g. refinement of requirements)

• Validation is to a great deal specification of requirements
  The rest is software engineering skill at different levels.

• A guide that is - as regards the importance - equivalent to GUM is necessary.
  (However: It would become much more complex!)