University in Ljubljana

Faculty for Electrical Engineering

Tanasko Tasić

Special aspects of validation of software in legal metrology

DOCTOR'S DEGREE THESIS

Ljubljana, 2009

University in Ljubljana

Faculty for Electrical Engineering

Tanasko Tasić

Special aspects of validation of software in legal metrology

DOCTOR'S DEGREE THESIS

Mentor: Prof. Dr. Janko Drnovšek

Co-mentor: Prof. Dr. Dieter Richter

Ljubljana, 2009

Statement:

I hereby declare that I have elaborated this doctor's degree thesis independently, under the guidance of the mentor, prof. dr. Janko Drnovšek and co-mentor, prof. dr. Dieter Richter.

Tanasko Tasić

Ljubljana, 2009-06-23

Table of Contents

ABSTRACT

The present thesis analyses, develops and evaluates new scientific approaches to the validation of software embedded in measuring instruments related to legal metrology. In parallel with the scientific and technical aspects, the legal metrology specifics and aspects are considered. Legal metrology is the entirety of technical and administrative procedures established in law by public authorities in order to guarantee the quality of measurements relating to health care, safety, fair trade and official controls. Particular attention and concern for software aspects in legal metrology was motivated due to numerous problems, e.g. detected attempts of fraud in measuring instruments’ software, mostly in measurements related to commercial transactions.

In order to be able to apply proper validation methods to legal metrological software applications, it is inevitable to know the inherent specifics of these applications. These specifics are related to:

- specifics of measurement technique (support for high accuracy, reliability, availability, security and repeatability of measurements),
- technological implementation (design of measurement systems using components from various suppliers with different technologies),
- reducing the cost of operation and maintenance,
- support for consumer protection (fraud protection),
- support of new participants in the process of legal metrology,
- support for remote legal verifications and inspection,
- compatibility with in-house built software components, and
- finally, supporting additional user functionality, which improves market competitiveness.

The theoretical and practical results of the work have been implemented in the guidance document WELMEC 7.2 "Software Guide (Measuring Instruments Directive 2004/22/EC)" [41]. In addition, a method for validation of applicability of such a guidance document has been developed and proven in practice through an experiment of comparative validation of the same instance of measuring instruments' software between several laboratories. This work may be considered as a pre-normative research in the standardisation of metrological software. The objective of the thesis is to provide particular general solutions – conformity assessment rules and procedures – not only for legal metrology but also for other technical systems where software plays a significant role.

The thesis is organised in eight sections.

Section 1 gives an overview of the scientific and related application (social) areas into which fits the explained topic. The thesis deals with the conformity of legal metrology measuring instruments to essential requirements of the related regulations, as assessed by conformity assessment bodies. At the

beginning, the role of metrology, and in particular legal metrology, in human life is explained. Being of extreme importance for human existence, metrology as a scientific discipline and infrastructural prerequisite ("conditio-sine-qua-non") is very well organised internationally. International metrological institutions (BIPM [50], OIML [51]), and regional (EURAMET [72], EURACHEM [73], EUROLAB [74], EA [75], WELMEC [52], COOMET [76], …) prepare procedural rules and supporting guidance documentation for assuring the traceability of measurements and conformity assessment of measuring instruments and the related applications. Requirements for the quality of metrological software may be regarded from several expert points of view, e.g. software technology standards, legislation, safety-related standards, or laboratory competence standards. The fulfilment of these requirements is confirmed by procedures called "conformity assessment", which may include testing, surveillance, inspection, auditing, certification, registration and accreditation. Within the contemporary "New Approach" to conformity assessment, responsibility for the quality of the measuring instrument is split between the manufacturer and conformity assessment authorities.

In Section 2 an overview of the impact of software on metrology is given. From the point of view of managing measuring systems, technological development brings many benefits: increased connectivity, facilitated design of larger, geographically wide-spread measurement systems, increased operability and maintainability (e.g. remote software update), enabled remote surveillance (inspection) functionality. The Internet enables completely or partly remote calibrations and web-enabled metrological services such as software validation tools. Nevertheless, the accessibility of widely spread metrological information has enabled a significant improvement in metrological knowledge. This progress is undoubtedly useful for the users, because it enables faster measurements, higher accuracy, and opens up the possibility of various analyses and further processing. For the manufacturer, the new technology simplifies the implementation of complex functions and gives them the flexibility to meet the wishes of their customers, and facilitates measuring instruments' maintenance. In such applications, there are various participants that need to have access, e.g. to an electricity meter, these being: end customers, various distributors, metering data provider, electricity supplier, owner of the distribution network, operator of the IT network, notified bodies, inspection bodies, legal verification authority, and of course, the meter manufacturer; each of them with their own access rights. For the electricity distribution companies, besides other benefits of remote meter operation (e.g. measurement data collection and dynamic tariffing), the highest reduction of costs comes from the possibility of remote updating of the software in measuring instruments already installed at the place of use. In the case of detection of a serious bug in a measuring instrument's software, maintenance costs are significantly lowered through remote software update. (The cost for a technician to visit an installation and physically download the new software might be even higher than the original meter retail price!) Taking into consideration the variety of parties involved, such system has to be adequately coordinated, monitored and supervised.

Section 3 gives an overview of software testing. Software testing is the most important part of software validation, which is necessary for the measuring instrument's conformity assessment. Certain software testing activities take place in each development phase of a software product. These phases are: user requirements analysis, functional requirements specification, technical requirements specification, design of software modules, and coding. The corresponding software testing activities are: module testing, integration testing, system testing, and acceptance testing. The basic software testing characteristics and approaches are described in this section, including: extent of testing, principles, techniques, strategies and organisational issues. It is explained why complete testing could practically not be achieved. Basic information on practical criteria for completion of testing is provided, as well as a brief overview of the available tools for automation of the testing process. In continuation, a short overview of the applicable software security testing methods is presented. The section ends with a short explanation of the specifics of testing the Internet applications and evaluation of the quality of the software (development) process. It is important to stress that the majority of errors remain hidden in the software code because of insufficient time for their fixation, due to the market pressure (to release the product before the competitors), and not because of a lack of programming skills, or malicious intentions of the programmers.

Quality Requirements for Metrological Software are presented in the Section 4. There are several guidance documents worldwide addressing various aspects of quality of scientific and legal metrological software applications. An interesting fact about the creation of these documents is that they were developed spontaneously and completely independently of one another. Although drawn up without a systematically prepared common foundation, all these documents follow two approaches. The first is intended for metrologists who develop metrological software applications by themselves, and therefore need more knowledge about certain software lifecycle activities (both building and verification of individual phases). The second approach covers the needs of the users of COTS[1]-like applications (developed by a third party) and conformity assessment bodies. Issues such as for example lifecycle activities (design and verification), validation guidance (up to the source code analysis), risk analysis, content of reports; security, virus protection and identification of software, are tackled differently by individual guides.

Section 5 describes the implementation of scientific contribution in the form of a newly developed software validation guidance document (which has evolved into the WELMEC Guide 7.2), its structure and anticipated use. The domain requirements (“essential requirements” of the Directive on Measuring Instruments) are taken as the starting point. Taking into consideration the technological realisation of measuring instruments, special requirement blocks intended for technological functionality have been prepared (for built-for-purpose or workstation-based instruments with the possibility of embedded functionalities of data transfer, data storage, software separation and download of software), followed by

[1] Commercial Off the Shelf

instrument-specific requirement blocks. As regards the environment of the intended use, risk classes have been proposed (affecting the rigour of requirements and validation). The application of such modular structure of requirements is supported by validation guidance and examples of acceptable solutions. Finally, the documentation of the performed validation is addressed, including both the legal metrology aspects (necessary explanations in type examination certificates) and standard laboratory output documents (e.g. test report).

Section 6 presents the interconnection between WELMEC Guide 7.2 and the related fields.

Section 7 presents the performance of the validation experiment during which the suitability of the developed scientific approach was proven. Validation of the guidance document was performed by comparative validation of the same instance of metrological software, which was performed simultaneously by six laboratories from national metrological institutes. The intention was to prove that the guidance document WELMEC Guide 7.2 meets its intended purpose. This section explains organisational and logistic issues, and provides an analysis of the outcomes (identified requirements, applied test methods, overview of fulfilment of the requirements), followed by an analysis of different judgements of nonconformity and the necessary resulting activities (basically improvements of the Guide). Besides testing the suitability of the WELMEC Guide 7.2 document, certain indicative parameters regarding the performance of the work came out as well, which may directly apply to analysing the abilities of particular laboratories.

Section 8 outlines the expected contributions to science. The first contribution is the development of a generic approach to software validation in legal metrology. Another important issue was ensuring that the newly developed guidance document fitted into all relevant technical and infrastructural areas, such as existing international software quality standards, software testing standards, normative documents relating to metrological software, and worldwide guidance documents relating to metrological software. Yet another aspect was the connection with conformity assessment in legal metrology and, on the other hand, software testing methods and strategies. A comparative software validation experiment was developed for the validation of the newly developed document. Last contributions regard the applicability of the developed experiment to intended validation, and its possible future applications to similar fields.

The key terminology used in this thesis is presented in Section 10.8

1 Introduction - Description of the specific field on which the original scientific contributions are focused

Validation of software in legal measuring instruments and systems is a part of measuring instruments' conformity assessment process. This section gives basic information on contents, approaches and institutions that perform conformity assessment.

1.1 Motivation for the work performed

During preparation of the Measuring Instruments Directive (MID) [4] it became obvious that a precise, unambiguous guidance for validation of the measuring instruments' software needed to be prepared. At that time several guidance documents related to metrological software existed (as explained in Section 4), but it appeared that none of them were suitable for validation of software in MID measuring instruments. Therefore, it was necessary to develop a new guidance based on new, generic and modular approach. With that intention the EU-funded GROWTH project MID-Software (under contract number G7RT-CT-2001-05064), was carried out from 2002 through 2004. The intention of the project was to reduce possible ambiguities over the interpretation of software requirements and establish mutual confidence in the results of software validation and testing between legal metrology instruments' conformity assessment bodies in the countries with implemented MID. The main objective was to elaborate specific software requirements based on general requirements as given in Annex I of the MID, and to harmonise the application of these requirements among the participating organisations, and to lay down the results in guiding documents. The domain-related expertise was assured through involving participants from national metrology institutes, notified bodies and manufacturers of measuring instruments – 16 members from 13 countries took part in the network. A complete list of participants is available in Attachment 10.3.

At the start of the project, the major dilemma regarding the performers of the validation was whether the metrologists would be skilled enough for validation of software, or it would be necessary to engage software and IT testing experts.

The most important element of software validation is software testing. The basic elements important to testing of metrological software are presented in Section 3. Conformity assessment of measuring instruments is traditionally performed by metrological experts, who are not experts in software testing. Preparation of proper testing procedures and selection of appropriate software testing methods and strategies seemed to be a too demanding task for them. At first glance, the solution for this gap appeared to be in leaving software testing to software testing experts. On the other hand, software testing experts are not familiar at all, either with the fundamental metrological principles or with the intended metrological application of measuring instruments (and their software). This is a significant deficiency for proper testing of metrological software – the result of such testing may be very poor regarding most significant metrological aspects of the software under test.

After thorough analysis within the project group it emerged that the best solution would be to prepare a guidance document containing unambiguous instructions for testing metrological software for the domain experts (metrologists).

The structure of the developed Software Validation Guide is presented in Section 5. The development and performance of the method for confirmation that the newly developed guide fits its intended purpose, and following cognitions are presented in Sections 6, 7 and 8.

1.2 Description of the specific field of science on which the original contributions are focused

The proposed thesis fits into the specific scientific area of metrology; i.e., it is dealing with the quality of software involved in legal metrology.

Metrology as a scientific discipline relates to the proper functioning of practically every activity in human life. Whether we are aware of it or not, every human action is connected with some measurement, starting with the wake-up time, measuring the amount of water spent in bathroom or kitchen, energy or gas for preparation of first morning coffee, velocity of our car while driving to the office, amount of petrol at the gas station, and many others, involving all actions during each and every day of our lives.

Legal metrology, as a specific metrology branch, treats measurements that are identified as needing supervision by the State and are regulated by law. This includes measurements in the areas of human and animal health, protection of the environment, consumer protection, general technical safety and proceedings before jurisdictional institutions. A precondition for the use of a measuring instrument or system intended for use in these areas is an examination of their suitability for the intended use. Activities aimed to determine, directly or indirectly, that a process, product, or service meets relevant standards and fulfils relevant requirements are called "conformity assessment", and are performed by conformity assessment bodies. Conformity assessment activities may include [10]:

- testing,
- surveillance,
- inspection,
- auditing,
- certification,
- registration and
- accreditation.

Legal metrology instruments are subject to several instances of conformity assessments during their lifecycle. The first conformity assessment activity is performed by the manufacturer of the measuring instrument who conducts type testing. Next (legal metrology) conformity assessment procedure is type approval[1] which is the prerequisite for the measuring instrument to obtain the status of legal metrology instrument. First verification is necessary before the first use, followed by regular and extraordinary verifications and inspection examinations.

Institutions that perform these conformity assessment procedures are both governmental and private bodies. The quality of their work is supervised by national accreditation institutions.

According to existing European legislation, the precondition for placing a product on the market is the fulfilment of essential requirements of all relevant technical directives and associated technical standards concerning the regulated area. In the case of measuring instruments treated by this thesis, the relevant directives are:

- Directive 2006/95/EC of the European Parliament and of the Council of 12 December 2006 on the harmonisation of the laws of Member States relating to electrical equipment designed for use within certain voltage limits (LVD) [68];
- Directive 2004/108/EC of the European Parliament and of the Council of 15 December 2004 on the approximation of the laws of the Member States relating to electromagnetic compatibility and repealing Directive 89/336/EEC (EMC) [69];
- Directive 2004/22/EC of the European Parliament and of the Council of 31 March 2004 on measuring instruments (MID) [4];
- Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS) [70].

Fulfilment of legally binding requirements is usually considered as a prerequisite for placing the product on the market that is obligatorily fulfilled and is not questionable. Being considered as presumably fulfilled, this aspect of measuring instruments' quality remains practically invisible to their users. Next issue that is usually taken for granted is the fulfilment of the requirements of technical specification.

Manufacturers of measuring instruments are aware that additional features, beyond the basic legal requirements, referring to safety, environment protection, fair trade and health protection, etc., namely quality, are the properties that influence the competitiveness of their product most. Responsibility for fulfilment of all quality characteristics (as presented in Table 1-1) remains with the manufacturer.

The quality of software is therefore split into essential requirements defined by "law", and added features for competitive purposes.

[1] In recent legislation documents this phase is called “Type examination”

Group of quality characteristics	Direct importance to users (and awareness)	Conformity assessment performed by	Surveillance
Enhanced functionality, usability, easiness of operation, maintainability	High	Manufacturer's laboratories, third-party laboratories	Market rules
Conformance to the technical specification	Taken for granted	Manufacturer's laboratories, third-party laboratories	Market rules
Essential requirements arising from legislation (safety, environmental, essential metrological)	Almost none, except in case of malfunction with serious consequences	Manufacturer's laboratories, third-party laboratories, notified bodies	Governmental structures (inspection bodies)

Table 1-1: Product quality characteristics

In some specific industrial areas (i.e. railway and automotive industry, aerospace industry, nuclear power industry and medical devices industry), software-related requirements have their roots in risk assessments of software malfunctions. These areas have already developed mature requirements, often with respect to software safety and reliability ([77-86]). This degree of maturity has not been reached in legal metrology. Failure in metrological software applications (in other areas besides the ones mentioned) might not lead to catastrophic losses or damage. But from the perspective of the hierarchical structure of metrology systems and consequently of leverage factor, damage may become tremendous (an illustrative example may be the health hazard caused by erroneous calibration of an entire population of biomedical measuring instruments because of inaccurate national standard).

The parties involved, who are directly interested in the quality of metrological software, are at least:

- users of measuring instruments,
- manufacturers of measuring instruments and systems,
- governmental institutions,
- conformity assessment authorities,
- conformity assessment laboratories,
- inspection bodies.

Requirements for the quality of metrological software come from the legislation, domain (metrology) standards, software technology standards, safety-related standards, IT security standards, laboratory competence standards, etc. Software conformity assessment is supported by several technical standards, guidance and normative documents [37]. There are two possible approaches to the assessment of metrological software quality [49]: product assessment (examination of the quality of software product) and process assessment (assessment of the activities in the software lifecycle).

The basic quality characteristics of software products are defined in the international standard ISO/IEC 9126-1:2001 Software engineering – Product quality – Part 1: Quality model [23]. These quality characteristics are: functionality, reliability, usability, efficiency, maintainability and portability.

On the other hand, an example of a legislative document that concerns metrological software is the EU Directive on Measuring Instruments (MID), which covers 10 groups of measuring instruments. In most cases, software related requirements are hidden in the context of the directives. A case study within this thesis is focused on validation of software of the measuring instruments covered by MID.

1.3 International metrological infrastructure

In order to enable our everyday functioning, all measurements need to be coordinated, from the level of common measurements for everyday use to the level of worldwide highest accuracy measurement standards. For this reason the international metrological community has established appropriate infrastructure. The worldwide cover metrology organisations are the International Bureau of Weights and Measures (BIPM), dealing with overall metrological issues, and the Organisation Internationale de Métrologie Légale (OIML) for legal metrology. The corresponding organisations in Europe are the European Association of National Metrology Institutes (EURAMET) and the European Legal Metrology Cooperation (WELMEC).

1.3.1 International Bureau of Weights and Measures (BIPM)[50]

The international metrology system is coordinated and managed by the cover organisation – International Bureau of Weights and Measures (Bureau international des poids et mesures – BIPM). It was set up by the Metre Convention and has its headquarters near Paris, France. It is financed jointly by its Member States and operates under the exclusive supervision of the International Committee for Weights and Measures (CIPM). Its mandate is to provide the basis for a single, coherent system of measurements throughout the world, traceable to the International System of Units (SI). This task takes many forms, e.g. definition and realisation of the units of measurement, preparation of guidance documents for supporting the international system of measurements, direct dissemination of units (as in the case of mass and time – e.g. UTC[2]), support to research in various fields of measurement science, coordination through international comparisons of national measurement standards (as in electromagnetic and ionizing radiation) and calibration of certain standards for countries – signatories to the Metre Convention. The BIPM has an international staff of over 70 and its status vis-à-vis the French Government is similar to that of other intergovernmental organisations based in Paris. The budget e.g. for 2008 was over ten million euros.

[2] Coordinated Universal Time

1.3.1.1 The Convention of the Metre

The Convention of the Metre (Convention du Mètre) is a treaty which gives authority to the General Conference on Weights and Measures (CGPM), the International Committee for Weights and Measures and the International Bureau of Weights and Measures to act in matters of world metrology, particularly concerning the demand for measurement standards of ever increasing accuracy, range and diversity, and the need to demonstrate equivalence between national measurement standards. The Convention was signed in Paris in 1875 by representatives of seventeen nations. As well as founding the BIPM and laying down the way in which the activities of the BIPM should be financed and managed, the Metre Convention established a permanent organisational structure for member governments to act in common accord on all matters relating to units of measurement.

The Convention, modified slightly in 1921, remains the basis of international agreement on units of measurement. There are now fifty-one Member States, including all the major industrialized countries.

1.3.1.2 The International System of Units (SI)

The 11th General Conference on Weights and Measures (1960) adopted the name Système International d'Unités (International System of Units, international abbreviation SI), for the recommended practical system of units of measurement. The 11th CGPM laid down rules for the prefixes, the derived units, and other matters. The base units are a choice of seven well-defined units which by convention are regarded as dimensionally independent: the metre, the kilogram, the second, the ampere, the kelvin, the mole, and the candela. Derived units are those formed by combining base units according to the algebraic relations linking the corresponding quantities.

The SI is not static but evolves to match the world's increasingly demanding requirements for measurement.

1.3.2 International organisations of legal metrology

National legal metrology institutions are organised in several international associations, for example in Europe – WELMEC (European Legal Metrology Cooperation) [52], or worldwide – OIML (Organisation Internationale de Métrologie Légale) [51]. The basic aim of these associations is to achieve harmonisation of legal metrology activities (e.g. requirements and procedures), facilitate exchange of experiences between member institutions, and to promote removal of barriers to trade in the field of measuring instruments, e.g. by assuring the necessary pre-conditions for mutual recognition of conformity assessment procedures (reports and certificates) and results for particular categories of measuring instruments between member countries.

1.3.2.1 WELMEC (European Legal Metrology Cooperation)

WELMEC is the European cooperation in the field of legal metrology. Its Members are representative national authorities responsible for legal metrology in the European Union and European Free Trade Association (EFTA) Member States. WELMEC was created by the signing of a Memorandum of

Understanding (MoU) at a meeting in Bern, in June 1990. The MoU is of an exclusively recommendatory nature, however, and does not in any way bind its signatories. WELMEC remains a free cooperation in which agreement is sought on a range of issues of mutual interest and wide importance.

In developing a strategy to achieve its declared objectives, WELMEC has forged a close link with the European Commission. In organising work at the level of the Commission services, WELMEC will be taken into consideration as effective means of cooperation between the Commission and national authorities in the field of legal metrology, which means that it will be included as partner in the process of preparing the European technical legislation.

There are two harmonised Directives of the European Parliament and the Council in the area of measuring instruments: Directive 90/384/EEC on non automatic weighing instruments [3] (NAWI Directive) and Directive 2004/22/EC on measuring instruments [4] (MID Directive). All countries – members of the European Union have to accept type evaluation certificates issued by any relevant notified body[3]. As supporting institution WELMEC needs to assure guidance documents necessary for equivalent performance of conformity assessment procedures between different notified bodies [42], [46]. Such documents explain both technical and organisational aspects of the specific conformity assessment procedures.

WELMEC is publishing a number of Guides to provide guidance to manufacturers of measuring instruments and to notified bodies responsible for conformity assessment of their products. The Guides are purely advisory documents and do not themselves impose any restrictions or additional technical requirements beyond those contained in relevant EC Directives. Alternative approaches may be acceptable, but the guidance provided in this document represents the agreed view of WELMEC as to the best practice to be followed. The guidance document elaborated in this thesis, which has been developed for the validation of software in measuring instruments covered by MID, has been adopted as WELMEC Guide 7.2.

1.3.2.2 OIML (Organisation Internationale de Métrologie Légale)

The International Organization of Legal Metrology (OIML) is an intergovernmental treaty organisation whose membership includes Member States, countries which participate actively in technical activities, and Corresponding Members, countries which join the OIML as observers. It was established in 1955 in order to promote the global harmonization of legal metrology procedures. Since that time, the OIML has developed a worldwide technical structure that provides its Members with metrological guidelines for the elaboration of national and regional requirements concerning the manufacture and use of measuring instruments for legal metrology applications.

[3] A notified body is an independent body appointed by an agency within one of the European countries, usually governmental, as being capable of performing the duties of a notified body as defined by the directives.

The OIML develops model regulations called International Recommendations (i. e. [31], [34]), which provide Members with an internationally agreed-upon basis for the establishment of national legislation on various categories of measuring instruments. Given the increasing role of national implementation of OIML guidelines, more and more manufacturers are referring to OIML International Recommendations to ensure that their products meet international specifications for metrological performance and testing. Cooperative agreements are established between the OIML and certain institutions, such as ISO [56] and IEC [57], with the objective of avoiding contradictory requirements; consequently, manufacturers and users of measuring instruments, testing laboratories, etc. may simultaneously apply OIML Publications and those of other institutions. The main elements of an International Recommendation are Scope, application and terminology; Metrological requirements; Technical requirements; Methods and equipment for testing and verifying conformity to requirements; and the Test report format.

Besides recommendations, OIML develops other, horizontally-oriented documents, which address issues important for several categories of instruments (e.g. reliability, software, …), terminology, infrastructural issues, etc. An example of such a horizontal document is the D-31 ”General requirements for software controlled measuring instruments” [30], a document treating software in measuring instruments.

The OIML Certificate System for Measuring Instruments provides the possibility for a manufacturer to obtain an OIML Certificate and a Test Report indicating that a given instrument type (pattern) complies with the requirements of the relevant OIML International Recommendations. Certificates are delivered by OIML Member States that have established one or several Issuing Authorities responsible for processing applications by manufacturers wishing to have their instrument types (patterns) certified. OIML Certificates are accepted by national metrology services on a voluntary basis and as the climate for mutual confidence and recognition of test results developed between OIML Members. The System serves to simplify the type (pattern) approval process for manufacturers and metrology authorities by eliminating costly duplication of application and test procedures.

To summarize, international organisations in the field of legal metrology on the one hand, aim at minimizing barriers to trade for measuring instruments (by enabling internationally recognisable technical conformity assessment), and on the other hand, try to assure internationally comparable surveillance of national metrology systems.

Slovenia is a member of both WELMEC and OIML and an associate member of the Metre Convention.

1.4 Conformity assessment – an overview

Conformity assessment is any activity to determine, directly or indirectly, that a process, product, or service fulfils relevant requirements specified by technical standards or equivalent documents. Conformity assessment activities may include testing, surveillance, inspection, auditing, certification, registration and accreditation [10].

The general objective of all conformity assessment activities is in creating a sufficiently high level of confidence between manufacturers (which perform type testing in their own or third-party laboratories) and national legal metrology authorities to accept documents (e.g. reports and certificates) related to the placing of these products on their markets.

Metrology and calibration provide the basic language for the measurements, which is fundamental to testing, while quality systems, certification and inspection procedures provide the final demonstration of quality, i.e. quality in the sense of conformity to product requirements. Reliable conformity assessment is based on traceable measurements and harmonised test procedures performed by competent organisations. That means that countries participating in mutual conformity assessment agreements have to establish a technical infrastructure which is constantly being advanced to keep pace with technological development.

There are always possible differences in opinions on the conformity of given products to specific requirements, as well as on the interpretation of technical specifications and their application to given products. Differences of opinion can arise between national surveillance authorities within the safeguard mechanism, or simply within the context of coordination meetings. Differences of opinion can also arise between national market surveillance authorities, manufacturers or notified bodies.

At present there are no means e.g. at the level of the European Union to contribute to the resolution of such disputes other than to organise meetings of experts, which, however, are not always conclusive. It is therefore suggested to examine the possibility of setting up inter-comparison of test results or proficiency testing programmes in order to support and strengthen mutual recognition of test results, conformity declarations and certificates. However, the possibility of setting up such a programme is directly dependent on available budgetary resources, and – more importantly – on management resources and procedures under the new financial regulations. Such a programme would not only contribute to the credibility of the certificates delivered, but also to that of the manufacturers and of the notified bodies, and could help to reinforce the effectiveness of accreditation.

It is important to point out that before placing a product on the market, it is necessary to ensure its conformity to every relevant directive or regulation, as explained in Section 1.2.

The procedures of assessing the conformity of a measuring instrument to different directives may vary largely. The present thesis addresses the aspects of the Directive on Measuring Instruments only.

According to the "New Approach" in the European Union, there are several conformity instances and possibilities [71]. An overview is presented in Table 1-2.

Module Designation	Short Description	Performer	Background (Examples)
A	Declaration of conformity based on internal production control	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005[4]
A1	Declaration of conformity based on internal production control plus product testing by a notified body	Manufacturer NB[5]	ISO 9001:2000 ISO/IEC 17025:2005
B	Type examination	NB	
C	Declaration of conformity to type based on internal production control	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005
C1	Declaration of conformity to type based on internal production control plus product testing by a notified body	Manufacturer NB	ISO 9001:2000 ISO/IEC 17025:2005
D	Declaration of conformity to type based on quality assurance of the production process	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005
D1	Declaration of conformity based on quality assurance of the production process	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005
E	Declaration of conformity to type based on quality assurance of final product inspection and testing	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005
E1	Declaration of conformity based on quality assurance of final product inspection and testing	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005
F	Declaration of conformity to type based on product verification	NB	ISO/IEC 17020:200x
F1	Declaration of conformity based on product verification	NB	ISO/IEC 17020:200x
G	Declaration of conformity based on unit verification	NB	ISO/IEC 17020:200x
H	Declaration of conformity based on full quality assurance	Manufacturer	ISO 9001:2000 ISO/IEC 17025:2005
H1	Declaration of conformity based on full quality assurance plus design examination	NB	

Table 1-2: Conformity assessment modules - "New Approach" [4]

In the case of an active electrical energy meter, which was considered in the case study of the present thesis, this means that:

– B+F: A representative of the type of an instrument needs to be examined by a NB. Each instrument unit from the serial production needs a legal verification by a NB, or

[4] The International Standard ISO/IEC 17025:2005 "General requirements for the competence of testing and calibration laboratories" is the standard defining system-level procedures for testing and calibration laboratories. Besides this one, the laboratories need to follow numerous "purely" technical standards for every particular aspect of measuring instrument or measuring process.

[5] Notified Body

- B+D: A representative of the type of an instrument needs to be examined by a NB. The manufacturer declares conformity to the approved type of each instrument from the serial production, based on his internal quality system, or
- H1: The manufacturer’s quality assurance system is approved and monitored by a NB. The design of each type of the instrument is examined and approved by a NB.

Several WELMEC guidance documents support the application of different conformity assessment modules [43], [44], [45] and [47]. The only document related to software conformity assessment is WELMEC Guide 7.2 [41].

In the case of the Directive on Measuring Instruments, the following combinations of modules are provided:

Annex	Measuring instruments	Possible combinations of modules
MI-001	Water meters	B+F or B+D or H1
MI-002	Gas meters and volume conversion devices	B+F or B+D or H1
MI-003	Active electrical energy meters	B+F or B+D or H1
MI-004	Heat meters	B+F or B+D or H1
MI-005	Measuring systems for the continuous and dynamic measurement of quantities of liquids other than water	B+F or B+D or H1 or G.
MI-006	Automatic weighing instruments	For mechanical systems: B+D or B+E or B+F or D1 or F1 or G or H1. For electromechanical instruments: B+D or B+E or B+F or G or H1. For electronic systems or systems incorporating SW B+D or B+F or G or H1
MI-007	Taximeters	B+F or B+D or H1
MI-008	Material measures	Material measure of length F 1 or D1 or B+D or H or G Capacity serving measures: A1 or F1 or D1 or E1 or B+E or B+D or H.
MI-009	Dimension measuring instruments	For mechanical or electromechanical instruments: F1 or E1 or D1 or B+F or B+E or B+D or H or H1 or G. For electronic instruments or instruments incorporating SW: B+F or B+D or H1 or G
MI-010	Exhaust gas analysers	B+F or B+D or H1

Table 1-3: Applicable conformity assessment modules for MID instruments [4]

2 Impact of information technology in metrology

Understanding the state-of-the-art of the technological realisation of metrological software applications is very important for the implementation of appropriate approach to their validation, e.g. for the refinement of (testable) requirements, determination of the risk class and selection of validation methods. Additionally, knowing the structure of applications may facilitate other conformity assessment processes, e.g. legal verification and metrological surveillance. The following section provides an overview of the field of metrological software and IT applications. Solutions from both legal and laboratory metrology are described, since experiences can be shared between both areas.

As said during the FASIT workshop [48], the development of legal measuring instruments began with the “iron age” (mechanical measuring instruments), continued through the “electronic age”, followed by the “software age”; and now we are already in the “communication age”. This statement is undoubtedly true for all areas of metrology that benefit from the state-of-the-art computer communication, web and database technologies.

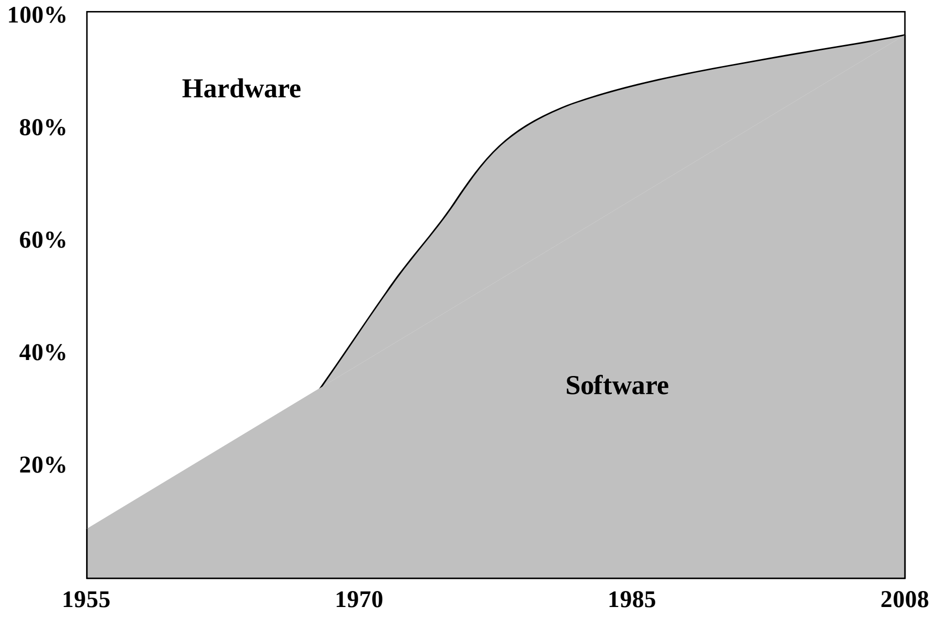


Figure 2-1: Trends in distribution of functionality between hardware and software in modern electronic devices, (modified from [87]]

As everywhere in technological development, software has become an indispensable part of metrology (as illustrated in Figure 2-1). Modern measuring instruments rely on embedded software, and data obtained through measurements are further processed by software systems. Besides the software running on a measuring instrument's computer, communications and database systems are implemented to enable distributed measuring systems, measuring data storage and their subsequent processing. In addition to acquisition, storage and processing of the measurement data, there are also the functionalities of maintenance of measuring systems, calibration and surveillance of the measuring instrument’s metrological status. Most of these services are available via public communications networks; others will be available very soon. This progress is undoubtedly useful for the users, because it enables faster measurements, higher accuracy, and opens up the possibility of various analyses and further processing. For the manufacturer, the new technology simplifies the implementation of complex functions, facilitates the measuring instrument's maintenance, and provides the necessary flexibility for meeting the wishes of their customers.

This development has led to increasingly complex metrological as well as supporting functionalities. This complexity has different faces: It may not only concern the functionality of a measuring instrument including self-checking facilities, or the distribution and transmission of measurement data, but also the method used for data analysis and, in particular, for the determination of measurement uncertainty. One can say that a new world of metrology has been opened up by software. However, the new software-based functionality and the complexity of metrological systems are not advantageous only. They have also raised new problems and questions to be answered. By no means should the risks involved with software be ignored. Software of insufficient quality may lead to malfunctions with unlimited consequences. Moreover, software must always be considered to be a target of accidental or even intentional corruption. Thus it is evident that horizontal subjects, such as software quality assurance and the security of software and measurement data, play an important role in metrology nowadays [67].

Validation of measuring instruments' and systems' software has become an inevitable part of their conformity assessment process. In principle, validation of software is a procedure typically based on international technical standards, and software testing techniques and strategies. In reality, software is always focused on specific applications. Specific applications require specific validation approaches. Practical experience indicates that the general, generic software testing approach is not appropriate. There is a specific need for appropriate, focused and straightforward technology to be solved in validation of specific applications. The software in legal metrology is such specific application . The specifics of software in legal metrology, as compared to general software, arise from the following requirements and facts:

- Support for high precision, accuracy, reliability, repeatability and reproducibility of measurements;
- Support for reliability, security and availability of measuring results;
- Variety of simultaneously applied engineering technologies;

- Facilitation of building measuring systems from components made by various vendors (which are not always compatible in functionality and communication);
- Minimisation of expenses of putting the instruments into operation and their subsequent maintenance (remote download of bug-fixed versions of software for measuring instruments in service);
- Support for proper consumer protection;
- Support for new parties involved in the operation of measurement systems (measurement data providers);
- Necessity of operation and validation support for software components developed by metrological domain experts (who are not software technology experts);
- Support to legal metrology verification procedures; and
- Support to metrological surveillance procedures.

2.1 Internet-enabled metrology [33]

State-of-the-art information technology enables a variety of specific metrological applications. An overview of the available solutions is provided in the following pages.

When mentioning internet-enabled metrology, people usually have in mind issues related to internet-enabled calibrations or websites with software validation tools. However, state-of-the-art IT enables a much wider variety of metrological applications. The benefits to metrology arising from the spread of the Internet may be divided into three groups. The first benefit for the metrology community lies in significantly increased remote functionality of measuring systems (e.g. connected in distributed legal metrology measuring systems, or remote operation of measuring instruments in severe environmental conditions). Next significant improvement comes from the introduction of new metrological services (e.g. time service, remote calibration and remote software validation). Finally, it is important to mention the increased availability of the metrology-related information, which is available on the World Wide Web.

The information technology employed for metrological applications covers more than public communication networks, as it is enabled by a range of hardware realisations and different operators such as: optical cable, cable TV networks, wireless, fixed-wire and mobile telephony (GSM[1], GPRS[2], UMTS[3]) or public networks using secure (e.g., VPN[4]) channels. In distributed measuring systems,

[1] Global System for Mobile communications

[2] General Packet Radio Service

[3] Universal Mobile Telecommunications System

especially in cases when distribution companies possess their own distribution networks (e.g. electricity), the common ways of data exchange are for example PLC (Power Line Carrier), DLC (Distribution Line Carrier), RADIAN (Radio Application Network) and ZigBee (IEEE 802.15.4:2006). Additionally, the supporting communication protocols may be general-purpose internet protocols (SMTP[5], HTTP[6], FTP[7]…) or application specific protocols, e.g. DLMS [15].

From the point of view of IT security, metrology-related IT applications apply standard approaches (e.g. HTTPS[8], FTPS[9], public key infrastructure), mostly for the protection of data (to ensure correctness of measuring data), and for the authentication of involved parties.

These approaches are not exclusively specific for the metrology community; however, it is important to mention them in order to make metrologists aware of their existence.

In the background, very often as the central point of a distributed measuring system, there is a database system. The applications again range from data collection/billing or dynamic tariff calculation (electricity) in legal metrology, medical diagnostics (databases of particular health states' pattern signals intended for medical diagnostics), general information about metrological capabilities (e.g. national), to monitoring & managing of distributed metrology systems (e.g. information about the bodies performing legal metrology tasks or information about the status of the measuring instruments, information about the instruments' calibration parameters during the lifecycle), and so on.

Metrology institutions worldwide are fully aware of the importance of these issues. The International Organization of Legal Metrology (OIML) organised a “Seminar on measuring instruments' software” as early as 1999. International Bureau of Weights and Measures (BIPM), and the leading national metrology institutes have organised a series of conferences on the impact of information technology in metrology:

- BIPM – NPL[10] Workshop on the Impact of Information Technology in Metrology, Teddington, UK, 16–19 September 2002,
- NMIJ[11] – BIPM Workshop on the Impact of Information Technology in Metrology, Tsukuba, Japan, 18–20 May 2005, and

[4] Virtual Private Network

[5] Simple Mail Transfer Protocol

[6] Hypertext Transfer Protocol

[7] File Transfer Protocol

[8] Hypertext Transfer Protocol Secure

[9] File Transfer Protocol Secure

[10] National Physical Laboratory (UK), http://www.npl.co.uk

[11] National Metrology Institute of Japan, http://www.nmij.jp/

- PTB[12] – BIPM Workshop on the Impact of Information Technology in Metrology, Berlin, Germany, 4–8 June 2007.

2.1.1 Functionality of measuring instruments (distributed measuring systems)

2.1.1.1 Legal metrology applications

Legal metrology instruments are subject to several instances of conformity assessments during their lifecycle. The fulfilment of the legal metrology requirements, which is checked during the initial conformity assessment procedure (e.g. type approval), is the prerequisite for a measuring instrument to obtain the status of a legal metrology instrument. First verification is necessary before the first use, followed by regular and extraordinary verifications and inspection examinations during the measuring instrument’s lifecycle.

Institutions that perform these conformity assessment procedures are both governmental and private bodies. In recent years, more and more conformity assessment tasks are being entrusted to private bodies. Irrespective of the trends towards privatisation of conformity assessment services and towards removing barriers to trade, it is required to maintain an adequate level of consumer protection.

The role of state institutions (which are responsible for the operation of national metrology systems) is becoming more and more focussed on the collection and analysis of reports on conformity assessment procedures (from private bodies performing e.g. verification of legal measuring instruments), and on occasional metrological spot checks of measuring instruments. As a consequence, huge amounts of data need to be exchanged between these institutions in order to maintain a suitable quality of the national legal metrology system. This leads to additional measuring instruments' functionality, which is nowadays mostly implemented by software and information technologies.

An illustrative example is the operation of electricity meters in the de-regulated energy market (as illustrated in Figure 2-2). In such a system it is necessary to ensure fair trade for all involved parties, who need to have access to the electricity meter, each of them with its own access rights. These parties are: end customer; several distributors; metering data provider; electricity supplier; owner of the distribution network; operator of the IT network; members of the legal metrology system, i.e.: the notified body (performing initial conformity assessment), legal verification authority (performing legal verifications) and the inspection body (usually independent body performing metrological surveillance of the measuring instruments in use). And, of course, the meter manufacturer and/or his representative need to have access for maintenance purposes as well.

[12] Physikalisch Technische Bundesanstalt (Germany), http://www.ptb.de

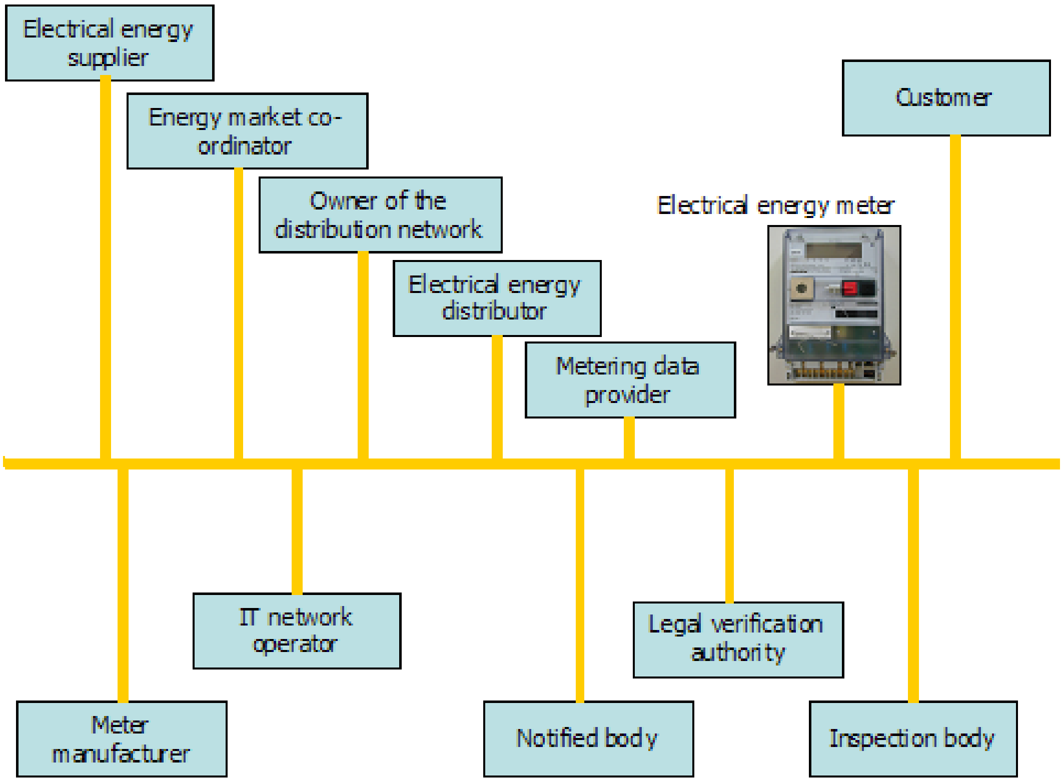

Electrical energy supplier
Energy market co-ordinator
Owner of the distribution network
Electrical energy distributor
Metering data provider
Electrical energy meter
Customer
IT network operator
Legal verification authority
Meter manufacturer
Notified body
Inspection body

Figure 2-2: Participants in a modern distributed measuring system – an example from legal metrology

The usefulness of remote access to meters is well illustrated by an example of operators of measuring instruments. For utility companies (e.g. electricity distribution companies) remote access significantly decreases the costs of collection of measurement data and maintenance of measuring instruments. The greatest decrease of operation costs is enabled by the possibility of remote update of software in measuring instruments already installed at the place of use. In the case of detection of a serious bug in the measuring instrument’s software, the software in all instruments of that type has to be updated. For electricity meters installed in remote mountain areas, the cost for a technician to visit an installation and physically download the new software, might well be more than the original meter's retail price! An acceptable procedure for remote update of the measuring instrument's software is proposed in the OIML document D 31:2008 “General requirements for software controlled measuring instruments”, as presented in Figure 2-3.

Taking into consideration the variety of parties involved, such a system has to be adequately coordinated. Authentication of every participant is necessary to ensure that only entitled personnel have access to particular meter data or functionality, and to make sure that the correct customer is charged for the consumption. Generally speaking, the risks of fraud in metrological software applications are not so high, compared to other areas (e.g. e-banking). However, they are certainly not negligible, as they vary from possible attempts to amend energy consumption data by market competitors, monitoring of energy consumption in order to detect absence of residents from home (e.g. by burglars), to just-for-fun hacker attacks.

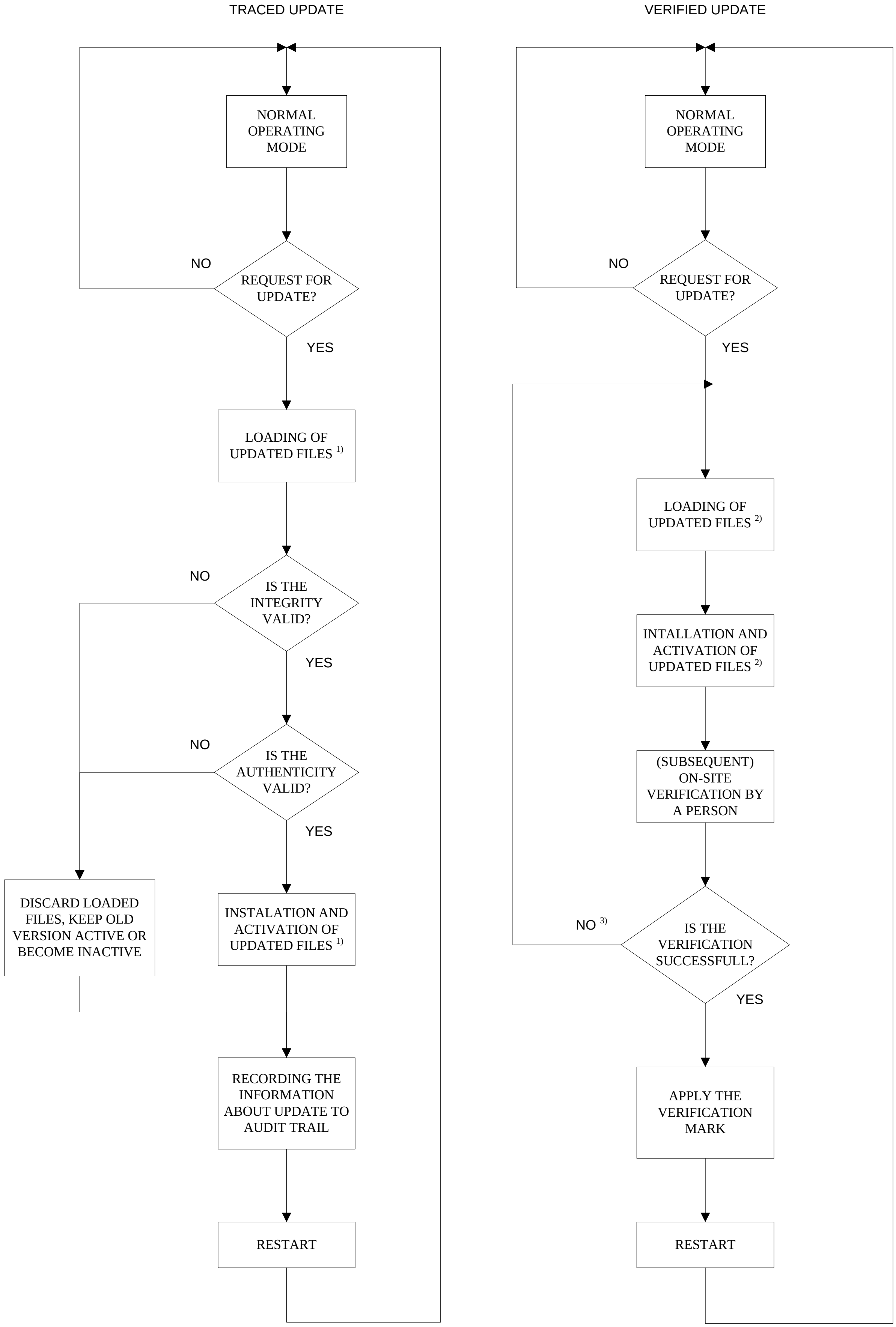

TRACED UPDATE
VERIFIED UPDATE
NORMAL OPERATING MODE
NO
REQUEST FOR UPDATE?
YES
LOADING OF UPDATED FILES 1)
NO
IS THE INTEGRITY VALID?
YES
NO
IS THE AUTHENTICITY VALID?
YES
DISCARD LOADED FILES, KEEP OLD VERSION ACTIVE OR BECOME INACTIVE
INSTALATION AND ACTIVATION OF UPDATED FILES 1)
RECORDING THE INFORMATION ABOUT UPDATE TO AUDIT TRAIL
RESTART
NORMAL OPERATING MODE
NO
REQUEST FOR UPDATE?
YES
LOADING OF UPDATED FILES 2)
INTALLATION AND ACTIVATION OF UPDATED FILES 2)
(SUBSEQUENT) ON-SITE VERIFICATION BY A PERSON
NO 3)
IS THE VERIFICATION SUCCESSFULL?
YES
APPLY THE VERIFICATION MARK
RESTART

Figure 2-3: OIML D-31: Software update procedure [30]

- Note 1: Traced update: updating is separated into two steps: “loading” and “installing/activating”. Software is temporarily stored after downloading without being activated, because it must be possible to discard the downloaded software and revert to the old version, if the checks fail.
- Note 2: Verified update: the software may also be downloaded and temporarily stored before installation, but depending on the technical solution, downloading and installation may also be accomplished in one step.
- Note 3: Only the failure of the verification due to software update is considered. Failure due to other reasons does not require re-loading and re-installing of the software, as indicated by the NO-branch.

Such system was developed, implemented and validated in practice during the SELMA [58] project ("Sicherer Elektronischer Messdaten-Austausch" – Secure electronic measurement data exchange). Participants in the project were manufacturers of measuring instruments, universities, electricity distribution companies and state institutes responsible for metrology and information security.

A similar approach has been implemented in road-traffic enforcement networks [24]. In such systems, automated vehicle velocity measuring stations (police radars) transmit files with evidence of offences to the place of processing. For the transmission of these files it is necessary to ensure their integrity and confidentiality, as well as proper authentication of participants in the process.

Being aware of the consequences of IT-related development in the field of measuring instruments, legal metrology institutions have prepared guidance documents that support harmonised software validation, such as the WELMEC Guide 7.2 "Software Guide (Measuring Instruments Directive 2004/22/EC)" and OIML D-31 "General requirements for software controlled measuring instruments". In particular, these documents include aspects of measurement data transmission, security of measurement data and download of software to measuring instrument.

2.1.1.2 Remote operation of measuring instruments

Remote operation of measuring instruments is important in circumstances where environmental conditions are such that human presence at the time and place of performance of measurements is not possible (e.g. in a mine, in a weather station, near steel production furnaces, in space vessels…), or in cases where permanent human presence is not practicable. There are many examples of the latter in industry (e.g. measurements after exposure of measuring instruments to stabilised environmental conditions or long-lasting measurements with periodic changes of reference values). It is important to stress that prerequisites for remote operation of measuring instruments are already present in the form of existing IT infrastructure.

An example of remote-controlled factory testing of an electric power monitoring system is illustrated in Figure 2-4: Several measuring instruments for monitoring the electric power parameters are connected in a computer network by different communication interfaces. Testing the functionality of measuring instruments comprises several long-lasting measurements. One measurement session of electric energy as a rule lasts several days. During that time, reference signals need to be changed several times. In addition, it is necessary to check the functioning of the system components (individual measuring instruments, measuring instruments server, database server, and client application). These checks need to be performed every few hours. Once initiated, all these checks can be conducted remotely – e.g. from home or a remote office– which is generally much more practical than visiting the testing laboratory every time some activity is necessary.

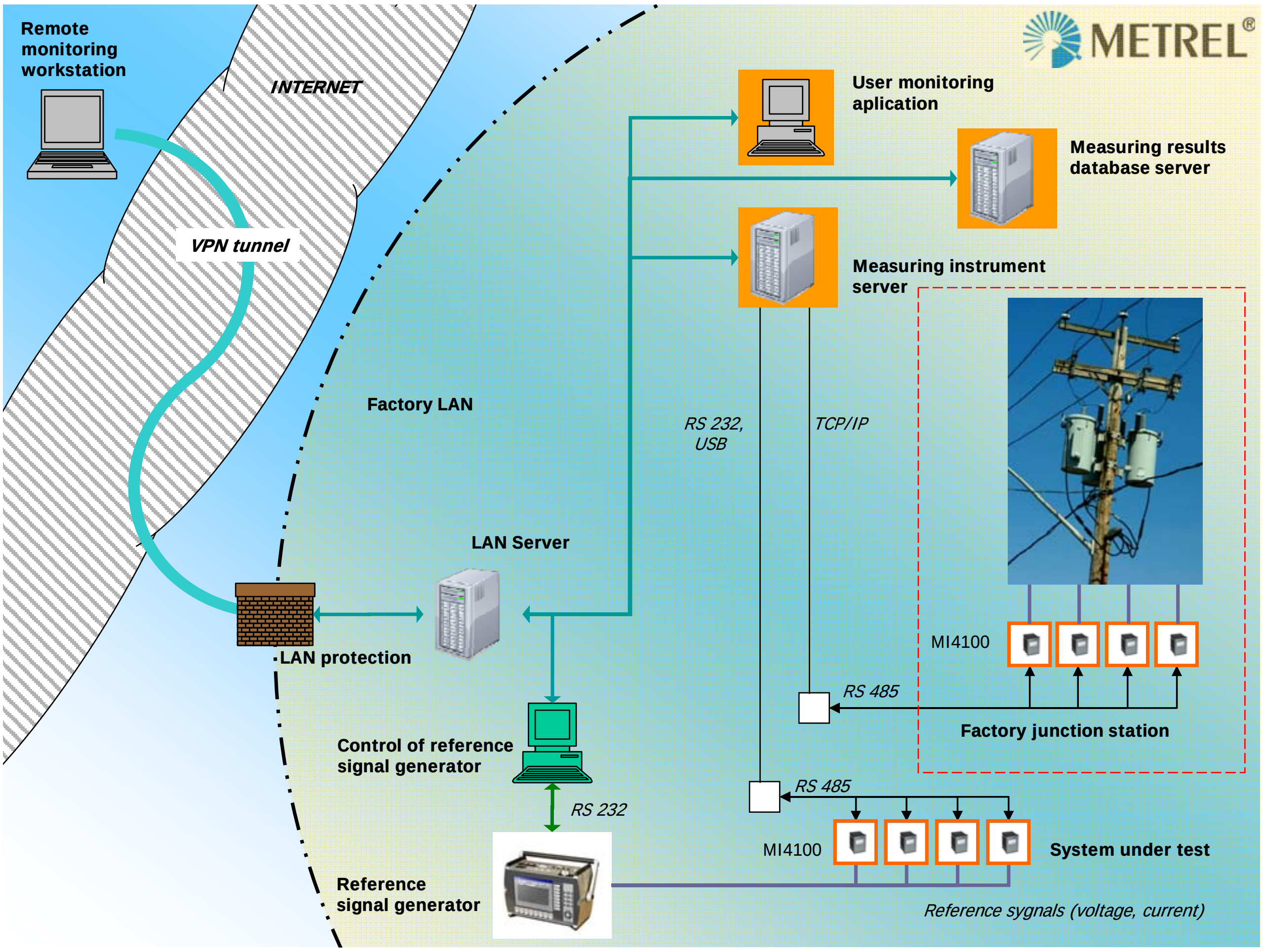

Remote monitoring workstation
INTERNET
VPN tunnel
METREL®
User monitoring aplication
Measuring results database server
Measuring instrument server
Factory LAN
RS 232, USB
TCP/IP
LAN Server
LAN protection
MI4100
RS 485
Factory junction station
Control of reference signal generator
RS 485
RS 232
MI4100
System under test
Reference signal generator
Reference sygnals (voltage, current)

Figure 2-4: A setup for remote testing of measuring instruments: Example from the company METREL, d.d.

Some of the software functionalities necessary for the support of remote measurements, are already embedded in operating systems (such as the Microsoft Windows™ virtual private network and remote desktop connections in the given example). Another solution is available in testing and measurement automation tools, e.g. publishing software control applications as web pages, which can then be remotely accessed (e.g. this functionality is available in the National Instruments LabVIEW™).

2.1.2 Internet-supported metrological services

Besides distributed measuring systems, which are intended for on-line uninterrupted service, there are very many Internet-supported services related to various metrological activities. These can be grouped as follows.

time requests from any Internet client by sending the time and estimated delay information in several formats including the DAYTIME, TIME (older realisations), and NTP protocols. The Network Time Protocol (NTP) is one of the most accurate and flexible means of sending time over the Internet. It can be used with almost any type of computer. The protocol is designed to compensate for some, but not all, network time delays between the server and the client. NTP is most successful across Local Area Networks, and can provide accuracy as good as a few milliseconds. On the World Wide Web, however, time transfer delays are not predictable, and accuracy figures cannot as easily be quoted. NTP conveniently supports security measures for users who want more reassurance concerning the origin of the time stamp (Authenticated NTP Services), rather than insecure NTP. At the client's side, the user needs to have software that can request time over the Internet. A version of the NTP client software used to synchronize computer clocks is called Simple Network Time Protocol (SNTP).

The accuracy of such services is rarely better than 0.1s, for better accuracies it is necessary to use other methods and services, e.g. Common-View GPS method.

2.1.2.2 Internet enabled/supported calibrations [59]

The "old-fashioned" procedure of performing calibrations, which is still used by the majority of customers and laboratories, involves transporting the instrument to be calibrated to the calibration laboratory. The instrument has to be physically present in the calibration laboratory during the entire calibration process. Such approach has several disadvantages, such as:

- long instrument downtime (for home laboratory),
- transportation costs,
- calibration of the instrument in conditions different from those of its routine use (e.g. other personnel, instrumentation environment, climatic environment), and
- risk of damage during transportation.

The spread of the Internet as a communication medium, and the availability of the continuously improved measurement standards, enable the dissemination of calibration values from higher-level standards laboratories in a different way. As a result, many institutions offer remote access over the Internet to more and more calibration and measurement services.

There are several ways in which the Internet can be used in support of the measurement process. From the point of view of logistics, the simplest way is there being no need to physically transport anything (no transfer standards). This way is possible for dissemination of a limited number of physical quantities (e.g., time, frequency), or in situations where measurement signal can be transmitted from the field laboratory to the reference laboratory. Other methods require either the presence of a reference standard (reference material), or the presence of a transfer standard at the place where the calibration is performed.

From the point of view of laboratories who need calibrations of their instruments, the benefits of Internet-enabled calibrations may be summarised as:

- reduced costs,
- direct access for more customers to the higher-level laboratories,
- the measurement standards belonging to the remote laboratory need not be transported,
- the standards are calibrated under normal conditions of use in their home laboratory,
- the "downtime" for the remote laboratory is kept at minimum,
- uncertainties are calculated on-line by reference software in the reference laboratory,
- the results can be reviewed on-line before the calibration is completed,
- instructions and procedures can be conveyed over the Internet,
- the calibration conditions can be recorded using a video or digital camera,
- the calibration environment can be recorded,
- the calibration chain is cut down to one link,
- calibration can be performed at any time; day or night, and
- the expertise of the higher-level laboratory can be transferred.

Some generic IT-related issues have to be addressed, namely:

- Security of calibration information and results as they are transferred over the Internet (data integrity, authentication, access control and protection from viruses and worms);
- Smooth operation of calibrations via the Internet through site fire-walls (regarding real-time),
- Overcoming the problems of two-way communication of data through fire-walls without compromising security;
- Division of software between web pages and compiled measurement code;
- Appropriate choice of computer-instrument interfaces at the remote site together with future proofing to accommodate evolution in computer software and hardware;
- Storage of calibration results in a database for recall by users during the calibrations, and the use of data warehousing to provide long-term access to calibration history;
- Guidance on the procedures necessary to meet accreditation requirements in general; and
- Promotion of the technology to those metrology areas where it may be applicable and advantageous, but has not been applied as yet.

For Internet-enabled metrology to be successful, it is essential that the software works reliably, both in terms of integrity of transmitted data and smoothness of operation from the point of view of the user. It is also necessary to prove that the traceability of a calibration when carried out at a remote site can be maintained. This is a non-trivial issue, as traceability covers factors such as suitability of the calibration environment, correct operation of the measurement equipment and, perhaps of most importance in this case, ability of the calibration staff at the remote laboratory to carry out the required tests under the guidance provided through web pages.

2.1.2.2.1 Entirely remote calibrations

An example of the entirely remote calibration is the use of Global Positioning System (GPS) satellite signals and Internet for time synchronization and frequency calibration [55, 60, 93], as presented in Figure 2-5. The satellite constellation consists of 24 satellites (the first was launched in 1978 and the 24th in 1994). They transmit signals that can be detected by receivers on the ground. The satellites are positioned in six Earth-centred orbital planes with four satellites in each plane. This means that 100 percent of the time, any point on the Earth's surface can receive signals from at least six satellites. The main application of the GPS system is the determination of the position of objects on the Earth's surface (primarily for military purposes). However, this is not the only application of the system. Each satellite carries either rubidium or caesium oscillators, or a combination of both, which is synchronised with UTC[13], USNO[14] and UTC NIST[15]. These UTCs are maintained within 100 ns of each other, and the frequency offset between the two time scales is less than 10^{-13}.

GPS receivers at locations A and B receive time information from the same satellite (which is in common view for both locations, A and B). The common-view method compares two clocks or oscillators located at different places. Unlike one-way measurements that compare a clock or oscillator to GPS, a common-view measurement compares two clocks or oscillators to each other. The first scientific publications describing this system, called "Common View", appeared as early as 1980. There are several types of time and frequency measurements that utilize GPS signals, some of them require intensive calculations that used to take up to several weeks in the past. By combining the common-view technique with the Internet, it is possible to build a common-view network, which processes data in near real-time.

The lowest uncertainties currently achieved using the GPS Measurement Techniques for Carrier-Phase Common-View are less than 500 ps for time and less than $5 \cdot 10^{-15}$ for frequency.

This approach may be the starting point for the realisation/dissemination of all physical quantities that are related to frequency (e.g. voltage).

[13] Coordinated Universal Time, http://www.bipm.org/

[14] The U.S. Naval Observatory, http://www.usno.navy.mil/

[15] National Institute of Standards and Technology (U.S.A.), http://www.nist.gov/

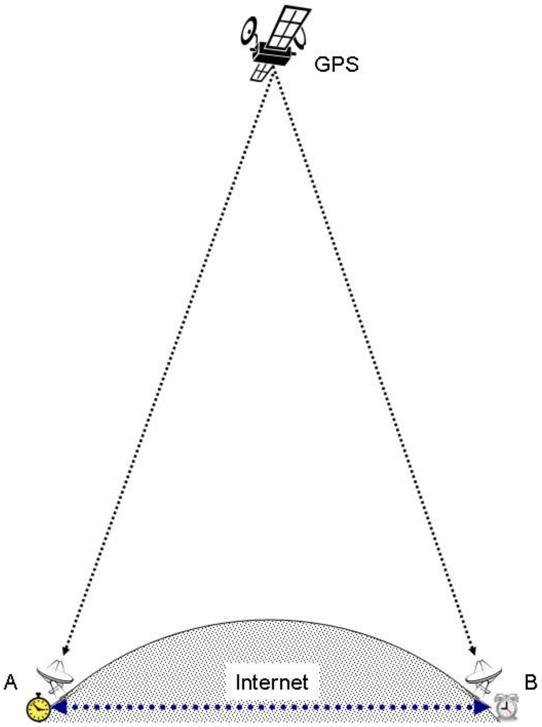


Figure 2-5: The Common View Method

2.1.2.2.2 Internet -– supported calibrations

There are several variations of Internet supporting calibrations. The main difference involves the necessity of presence of a local reference standard at the place of performing the calibration. With regard to that, we distinguish:

- Internet-supported calibration with transmission of measurement signal to reference laboratory,
- Internet-supported calibrations with a local reference standard, and
- Internet-supported calibrations with a travelling reference standard.

The first variation (presented in Figure 2-6) will be explained on an example of calibration of length standards [62]. The laboratory requiring calibration has its own low-coherence interferometer (LabI), and performs measurements on a unit under calibration (UUC). The measurement signal, in the form of the interferometer signal, is transmitted via optical fibre to the reference laboratory. The reference laboratory with its associated software then determines the parameters of the UUC. The main drawback of this method is signal loss in optical fibre between the customer premises and the reference laboratory. With the state-of-the art technology, affordable distance is ca. 20 km.

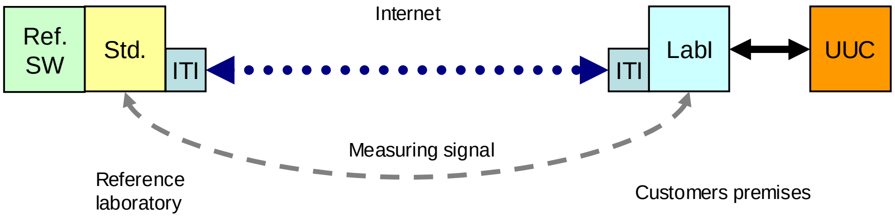


Figure 2-6: Internet-supported calibrations with additional transmission of the measurement signal

ITI -— IT Interface
UUC —-Unit Under Calibration
LabI -— Instrument in customer's laboratory – here used as the light source
Std. -— Higher-level measurement standard
Ref. SW – Software associated with higher-level measurement standard

Internet-supported calibration with the local reference standard (Figure 2-7) uses a local reference standard or material (Ref S) with stable, known characteristics for the calibration of an instrument under calibration (IUC) [61]. The IUC measures the characteristics of the Ref S and sends the results via the Internet to the reference laboratory. After analysis of the measuring results, the software in the reference laboratory calculates the calibration parameters for the IUC. This approach is used for the calibration of instruments in many areas, such as the measurement of impedance (iPIMMS, Primary Impedance Measurement Software for Impedance Calibration of Vector Network Analysers), spectrophotometry (iColour Calibration Visible Diode Array Spectrophotometer), pressure measurement, or ionising radiation measurement.

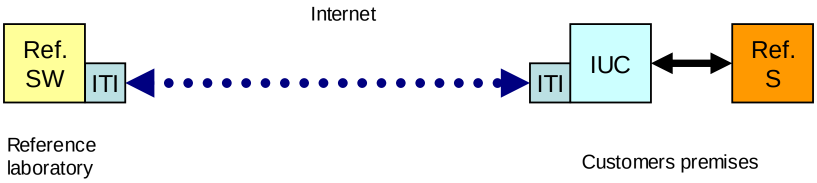


Figure 2-7: Internet-supported calibrations with the local reference standard

ITI —- IT Interface
Ref. S --— Reference sample (material)
IUC --— Instrument Under Calibration (in customer's laboratory)
Ref. SW --— Reference laboratory's software

Internet-supported calibrations with travelling reference standard (Figure 2-8) is applied in areas where local reference materials or reference standards of appropriate quality are not available. The transportation of travelling standards from the reference laboratory to the customer’s laboratory is generally more convenient than the transportation of the instrument to the reference laboratory. The related costs are likely to be lower as well. The procedure of calibration is the same as in the case of Internet-supported calibrations with local reference standard.

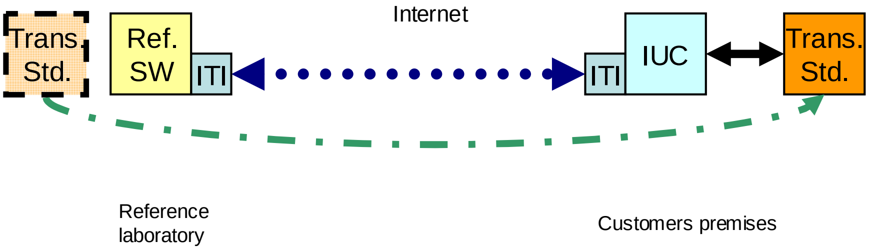


Figure 2-8: Internet – supported calibrations with travelling reference standard

ITI -— IT Interface
Trans. Std. -— Travelling reference standard
IUC -— Instrument Under Calibration (in customer's laboratory)
Ref. SW -— Reference laboratory's software

Travelling reference standards are not yet available with very high precisions; the uncertainties for this type of calibration are usually higher than the uncertainties obtained in reference laboratories.

Internet-supported calibrations may be of great interest for the legal metrology (conformity assessment), because very similar approaches can be used by manufacturers, notified bodies and metrological surveillance bodies for remote calibration, verification and surveillance of legal measuring instruments.

2.1.2.3 Availability of specific metrological software validation services

Modern laboratory measurement systems are very often highly automated in data acquisition as well as in post-processing of measured data [92]. Typically, the software components of such systems may consist of both commercially available software packages and custom-made software developed by the laboratory staff. In order to ensure confidence in the results provided by such systems, the software must be proven to be fit for purpose, so it must be possible to properly validate it, in whole and partially.

Web-based software validation tools make it possible for different users worldwide (developers, evaluators or laboratories) to validate their software modules using black-box testing methods [63, 64, 90].

Two approaches to validation of software are identified as appropriate for web-based software validation tools, namely: validation with the reference data sets, and validation with the reference software.

The following are examples of available tools :

- Web pages with reference software for validation of mathematical functions;
- Web pages with reference software for validation of standardised metrological functions; and
- Web pages with reference datasets for validation of metrological software.

The reference software for validation of mathematical functions enables the testing of software components, which realise standard mathematical or statistical functions (e.g. mean, standard deviation, square root, …). The user validates his software by comparing the output parameters from his and the reference software in response to the same input parameters. The reference software may be realised as a web application, or downloadable components that run on the client's machine.

The reference software for validation of standardised metrological functions is suitable for validation of software components, which realise algorithms based on well-known international technical standards (e.g. IEC 60584-1, 2 Thermocouples). Its purpose is to enable the developers (metrologists from laboratories that develop their own software components) to validate their software components. The application is simple; the user just needs to enter the input variables simultaneously into both the software under test and the reference software, and compare the results from both afterwards. Such application also allows laboratories involved in an inter-comparison to check their software modules before the beginning of the actual measurements, and thus eliminate any potential source of difference due to software errors.

Some additional issues arise in connection with this approach. The first concerns the validation of the reference software [65]. An application that claims to be the reference application needs to be suitably validated. Next issue relates to IT's security – the user needs to be 100% sure that the results come from the genuine reference software, and not from some fake (malicious) copy. This issue can be resolved with the use of appropriate Internet protocols (e.g. https), depending on the risk estimation. The software characteristic, which is usually validated using this approach, is functionality (numerical correctness), one of the most important characteristics for metrologists.

Instead of dynamically generated reference data, the reference datasets may be static, intended for validation of standardised metrological functions. The reference datasets must contain both input and output parameters. During the testing of his software, the user applies the input parameters, runs his software, and then compares the results with the reference one(s).

These services may be applicable in legal metrology for verification purposes.

2.1.2.4 Availability of the metrology-related data

Ease of distribution of information is the most important advantage of the Internet. In the metrological community, the most important information is provided by web-hosted databases containing relevant metrological information, such as:

- Calibration/metrological resources and their capabilities. The most comprehensive database collection containing worldwide metrological information is available on the website of BIPM. It contains information about scientific work, history and technical realisation of the primary measurement standards, calibration and measurement capabilities in particular countries, measurement units, technical committees, measurement standards, publications, conferences, key comparisons, reference materials and other metrological information;

- Research-oriented databases, e.g. PTB databases for vacuum-metrology or for medical research (electrocardiograms) [91];
- Information exchange centres, e.g. "virtual institutes". These sites are intended for experts in particular areas of metrology. Some examples are: Virtual Institute for Reference Materials (VIRM, http://www.virm.net/), Virtual Institute for Thermal Metrology [99] (Evitherm, www.evitherm.org/) and Virtual Institute of Energy Metrology (https://bi.offis.de/viem/tiki-index.php);
- Legal metrology databases are intended mostly for public information about nationally or internationally (EU Type Approval or OIML Type Approval Certificates) approved measuring instruments, metrological standards and regulations, ongoing projects, etc.;
- Metrology-related public information – information about approved measuring instruments, about their intended use, about control bodies and inspections;
- Databases of national metrology institutes contain a variety of data: organisational, calibration and verification capabilities, scientific background for metrological procedures, presentation of achievements, metrological advice for the general public, etc.;
- Databases of reference materials or reference data, e.g. at the Institute for Reference Materials and Measurements.

Various database technologies are employed in the design of metrological database systems. The choice of technology depends mostly on the amount of data in the databases, and the available resources.

Metrological databases may be used as a tool for monitoring the processes in a distributed metrology system. An example of such web-based database system (Figure 2-9) is realised in the Slovenian National Metrology Institute (MIRS) [66]. The implementation is adjusted to the specifics of the organisation of Slovenian metrology. MIRS is responsible for the entire scope of national metrological activities, including maintaining the system of national and reference standards for physical quantities and chemical measurements, the legal metrology system (type approvals, verifications, precious metals), metrological surveillance of legally controlled instruments and other issues, including the Slovenian Business Excellence Prize.

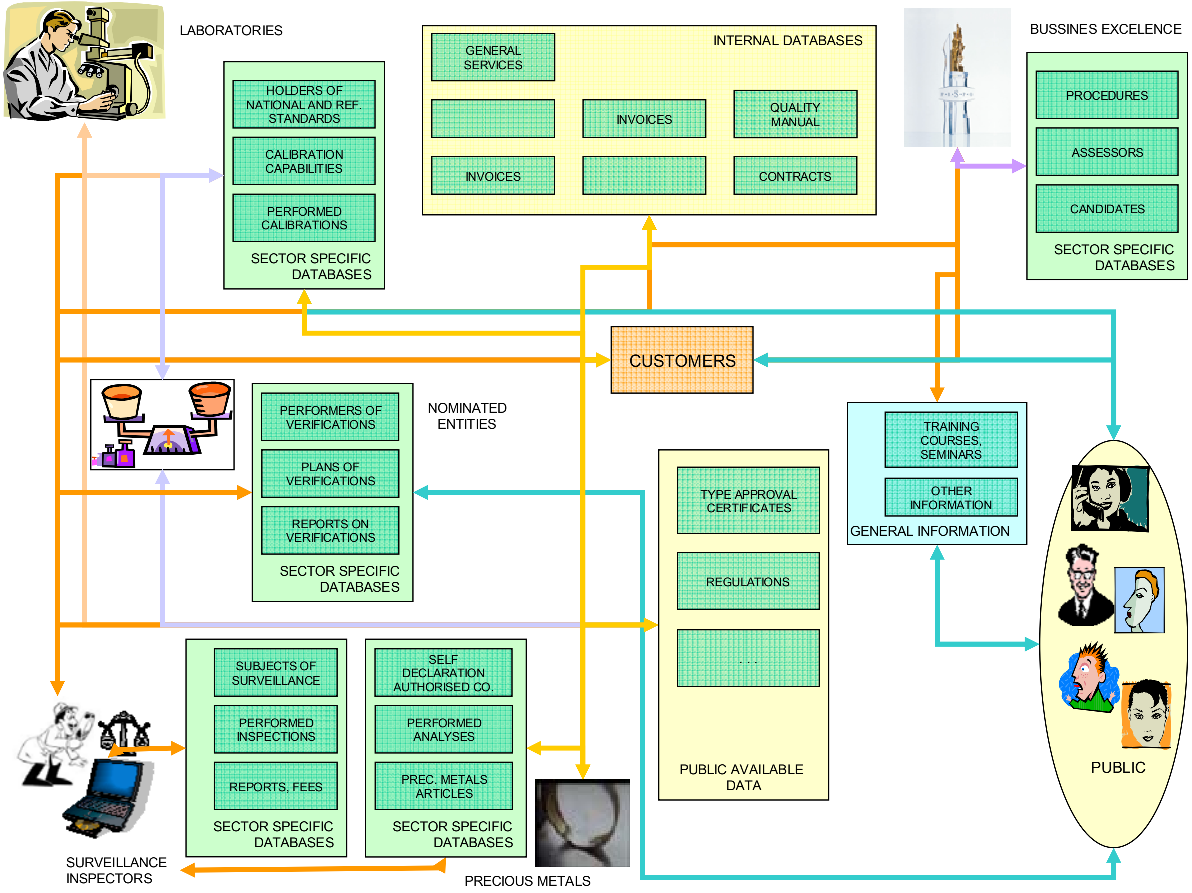


Figure 2-9: System of databases for monitoring the functioning of national metrology system – an example

Monitoring of all processes in such system requires acquiring, manipulating and processing a large amount of data. The MIRS implementation is suitable for monitoring the activities of a distributed metrology system in a small country, covering various aspects from the point of view of the responsible organisation, players, public and international partners. Besides facilitating the organisational issues, it provides transparent information to all members of the metrology community, from high-end metrological laboratories to the users of measuring instruments. The implementation is based on Linux[16] operating system with an Apache[17] server and MySQL[18]. Access and applications are realised through PHP[19] and JavaScript[20] scripts. Through the selected open-source technology it is possible to minimise the initial expenses for building such a system. If necessary, the platform may be changed afterwards, when the basic concepts have been clarified.

[16] http://www.linux.org/

[17] http://httpd.apache.org/

[18] http://www.mysql.com/

[19] http://www.php.net/

[20] http://javascript.internet.com/

2.1.3 Conclusions

The benefits arising from the new functionalities enabled by software and the spread of IT technologies, are manifold for the metrology community. Improvements originating from the application of the new software technologies (algorithms, optimization of compilers, graphical user interfaces) reflect in the improvement of metrological functionality – e.g. enhanced, faster and more reliable operation, higher accuracies and lower measurement uncertainties. The legal metrology area benefits from the functionalities such as storage and security of measuring results and reproducibility of measurements. The spread of the Internet facilitates the implementation of new or improved functionality of distributed measuring systems, such as remote operation (readout, in-field maintenance, calibrations, legal verifications, remote validation tools, remote metrological surveillance). The benefits consist in both fundamental improvement in some areas (achieving lower uncertainties, as in the case of time and frequency) and improved quality (functionality, reliability and faster performance) of metrological services. Increased availability of metrology-related information is very important as well.

In order to take full advantage of the new possibilities, they have to be implemented carefully (taking into account e.g. IT security issues) [97], with full understanding of the background of applied technologies. Both the conformity assessment and metrological software validation area need to keep in-line with the new developments.

3 Validation of metrological software (an overview of the activities of software quality assurance) [67], [49]

Software validation is defined as confirming – by providing objective evidence – that the requirements for a specific intended use or application have been fulfilled. The core validation activity consists in testing the software for confirmation of its suitability for intended use.

The basic software testing prerequisites applicable to validation of metrological software applications are explained in this section. The presented principles, techniques, methods and strategies are suitable not only for conformity assessment within the scope of legal metrology, but also for validation of metrological applications in general – also for metrologists who develop metrological software applications by themselves. Moreover, some deeper insight into software testing is given for better understanding of the technology background.

These activities are spread throughout the lifecycle of a software product; some of them occur during the development phases (verification of the particular development phase), others are performed on a finalised software product (validation).

To validate software as an integral part of a measuring system, different approaches of software quality assurance have to be applied for different conformity assessment procedures. There are two essential categories of software quality assurance, which supplement each other. On the one hand, the analytical methods of software testing, static analysis and code inspection are used within the scope of conformity assessments of final or intermediate products (suitable for use within the conformity assessment module “B”, as explained in Section 1.4). On the other hand, preventive audits of software development processes are applied to evaluate and improve appropriate software processes, and to consequently support process-related conformity assessment procedures (suitable for application of the conformity assessment module “H1”). Depending on the validation objectives, validation methods, audit areas, and the appropriate requirements have to be selected and refined. The process of defining and refining testable requirements is always a major issue of validation efforts

3.1 Software lifecycle

Software, either stand-alone or embedded into a device or system, has its own life cycle, similarly as any other industrial product. Several testing instances occur during the lifecycle of a software product. The selection of approaches to testing somehow depends on the lifecycle phase.

To understand the processes related to software quality assessment, it is necessary to identify the phases of a software lifecycle and the related software testing activities.

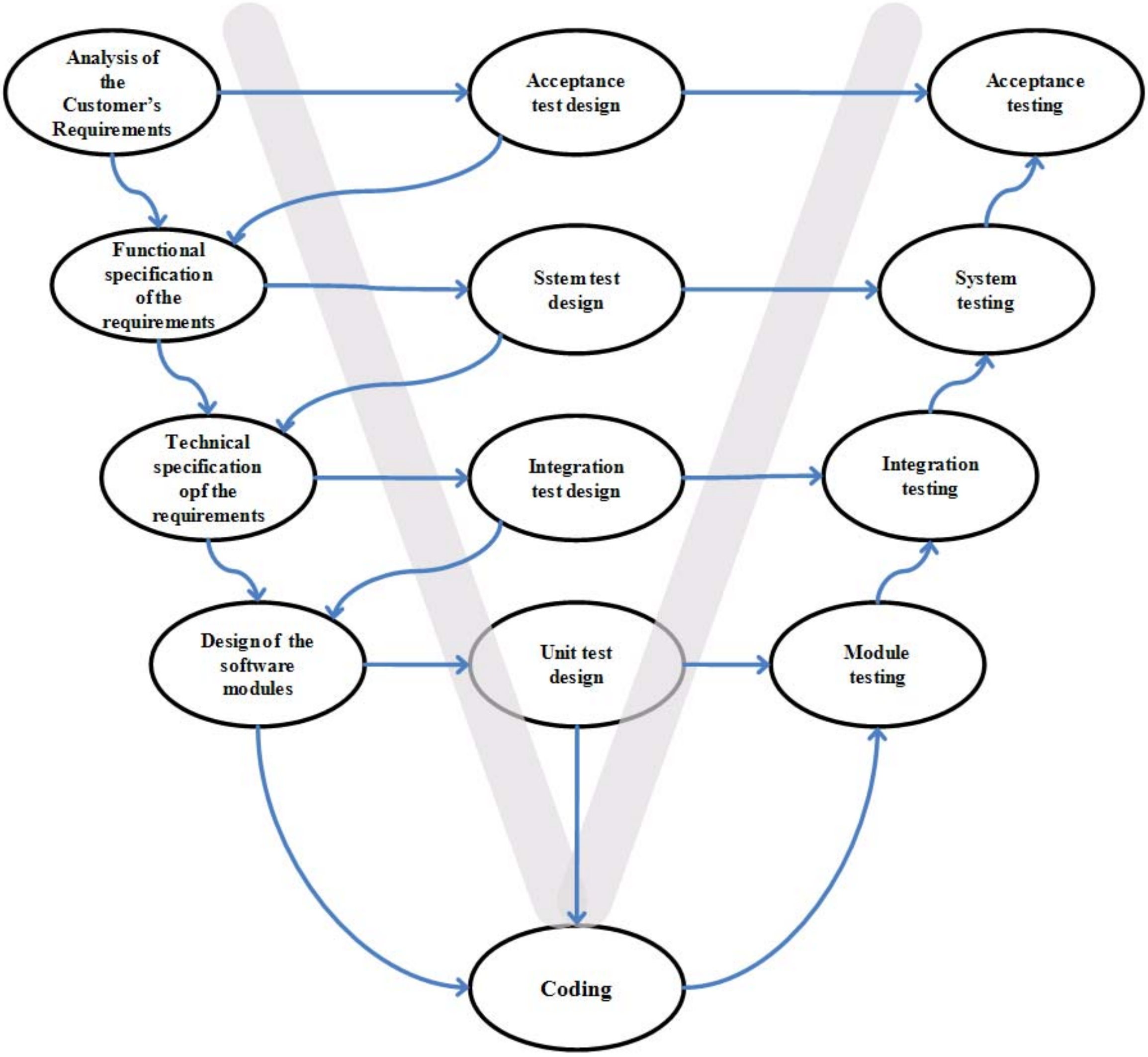

Analysis of the Customer's Requirements
Acceptance test design
Acceptance testing
Functional specification of the requirements
Sstem test design
System testing
Technical specification opf the requirements
Integration test design
Integration testing
Design of the software modules
Unit test design
Module testing
Coding

Figure 3-1: Software development lifecycle – V model

There are several ways of visualisation of the software development lifecycle [32]. Different authors split the software lifecycle into a different number of stages. The essential phases for understanding the issue are as follows:

- Analysis of customer's requirements.
- Functional specification of the requirements.
- Technical specification of the requirements. The functions and characteristics of the program should be detailed and documented. All the boundary conditions should be considered (hardware, operating system, users, and load) as well as the current and future needs (extension), the time, the financial and administrative capacity, and the target program reliability. The result of the analysis must be written in the form of system (program) specifications.
- Design of software modules. The design of the phase which, depending on the technology hardware and software, defines the structure of the software, data structures and communications, and working algorithms.
- Coding as the implementation of the design in software modules.

Each of these phases has its own corresponding testing phase, as presented in Figure 3-1. Details on software testing phases are given in the continuation of this section.

The software comes into operation after the development is concluded (after the acceptance testing is successfully concluded).

A very important phase that must not be neglected is the software maintenance, which is normally performed after the software product is already in use. There are several reasons for maintenance activities:

- elimination of errors identified during use,
- adapting the software to the requirements of the environment (for example, new hardware or new operating system),
- new customer requirements, and
- preventive maintenance.

Maintenance requires changes to specifications, additional programming and further testing – i.e., confirmation that the old functionality of the program is unchanged, and that the new functionality is properly implemented.

3.2 Software product testing [28]

Software faults occur through the following process:

A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defective code is executed, in certain situations the system will produce wrong results, causing a failure. Not all defects will necessarily result in failures (e.g. defects in “dead “code). A defect can turn into a failure when the environment is changed. Examples of these changes in the environment include the software being run on a new hardware platform, alterations in source data or interacting with different software. A single defect may result in a wide range of failure symptoms.

The definition of the software testing from the standpoint of software quality assurance is "reviewing and execution of a program to detect errors". On the other hand, from the user's point of view, software testing is "determination of compliance with the requirements" (conformity assessment).

The testing of software is not meant to confirm that there are no errors in the program. A program, which is functioning properly, may still contain errors. In a typical software product development cycle the share of the resources spent on testing is approximately 50% of the time, effort and cost. Because of this, testing is an expensive, time-consuming activity, which requires significant human and material resources. Therefore, the expected result of software testing is increased quality of the software, especially reliability. Increasing the reliability of the program means detecting and correcting as many errors as possible. Considering the specifics of the software technology, software testing can never be complete.

3.2.1 Characteristics of software testing

Taking into account the definition, one can say that software testing is a destructive process, as opposed to development, which is a constructive process. Only the test that causes a failure (discovers a bug) is a successful test case.

As a second subjective (psychological) moment, it is important to define realistic (achievable) objectives for testing. Solving crosswords is a good example. If someone is assigned to solve a complex crossword within fifteen minutes, very little or nothing will be solved after ten minutes. On the other hand, if one is given four hours to solve a crossword, much more will be solved after ten minutes. Man works much better when he is aware that the assigned objectives are achievable [28].

3.2.2 Extent, methods and strategies of software testing

An essential feature of software testing is its extent, which is directly related to the duration and cost of testing. There are different criteria for determining how to test software, and deciding when the program is tested sufficiently. As a rule, regardless of the extent of testing, all errors in a program will never be discovered.

The extent of testing and the extent of eliminating the errors are business decisions. Taking for example two competing manufacturers, who both develop a new version of their software products, the time to issue a new version is a very important parameter for financial success of the company [102]. Therefore, the extent of testing is time-limited, and the number of eliminated errors is consequently limited, too. It often happens that a company releases a product to market although being aware that it contains certain known errors. Such a decision is the result of the company's estimation of the balance between possible damage because of releasing a slightly erroneous product on the market, and loss of the market share because of late release of the product. Usually in such cases, the errors that cause frequent and obvious failures are removed, while other errors remain (with the intention to remove them in the next release of the product).

When considering an example from the software testing "Bible" [28] – a simple program called "triangle", which on the basis of the length of triangle sides determines the type of the triangle (whether it is an equilateral, isosceles, or scalene triangle), and calculates the sizes of the triangle's angles and the triangle's surface. A flowchart of the program is presented in Figure 3-2. It is a relatively simple program, in which there are no loops (neither iterative nor ordinary repeated loops). For proper testing it is necessary to test all segments of the code and all the decision points. This can be done by selection of the appropriate values of the test case's input parameters. In practice, there are two well-established strategies of testing – functional and structural testing.

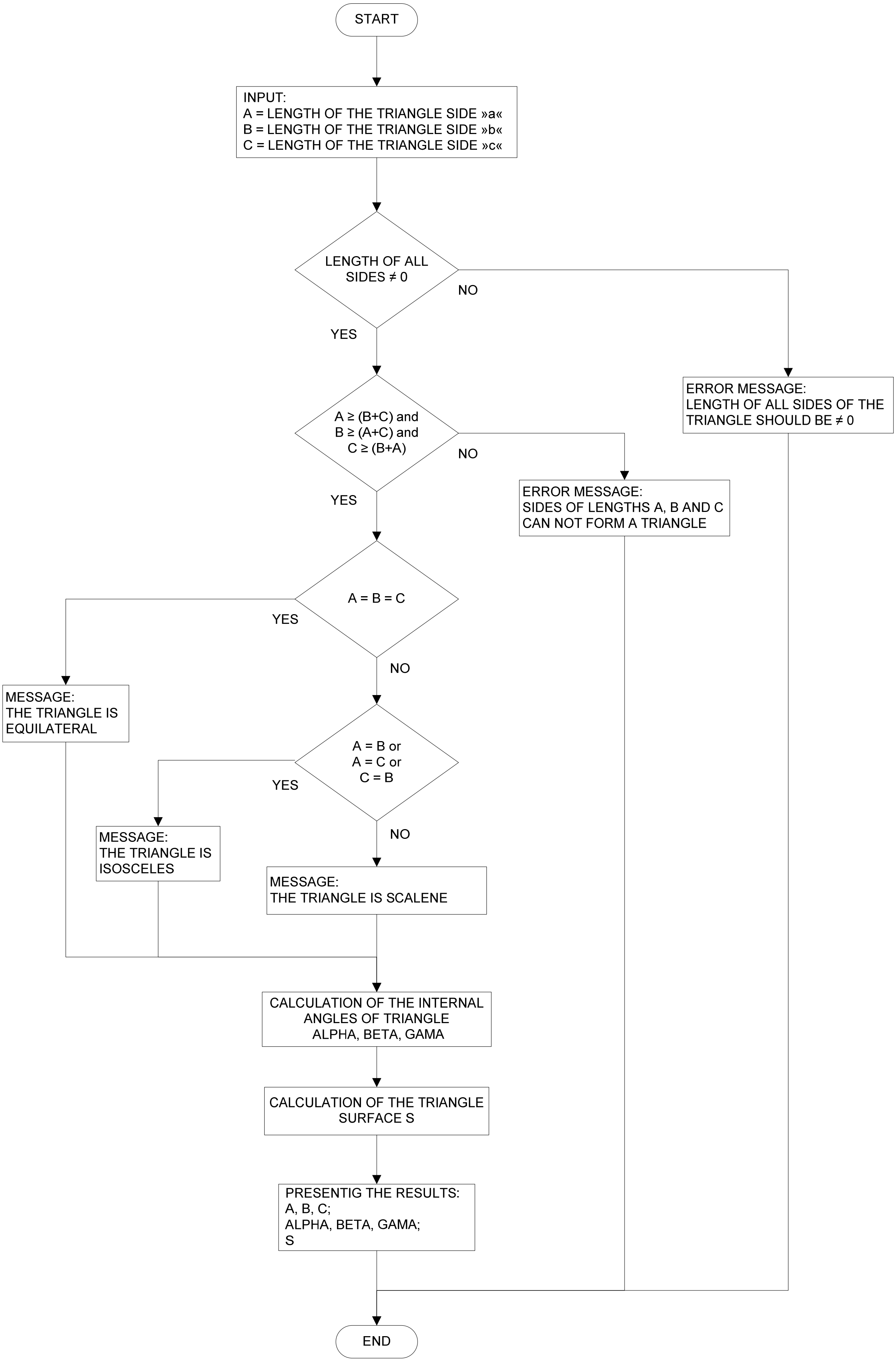

START
INPUT:
A = LENGTH OF THE TRIANGLE SIDE »a«
B = LENGTH OF THE TRIANGLE SIDE »b«
C = LENGTH OF THE TRIANGLE SIDE »c«
LENGTH OF ALL SIDES ≠ 0
NO
YES
ERROR MESSAGE:
LENGTH OF ALL SIDES OF THE TRIANGLE SHOULD BE ≠ 0
A ≥ (B+C) and B ≥ (A+C) and C ≥ (B+A)
NO
YES
ERROR MESSAGE:
SIDES OF LENGTHS A, B AND C CAN NOT FORM A TRIANGLE
A = B = C
YES
NO
MESSAGE:
THE TRIANGLE IS EQUILATERAL
A = B or A = C or C = B
YES
NO
MESSAGE:
THE TRIANGLE IS ISOSCELES
MESSAGE:
THE TRIANGLE IS SCALENE
CALCULATION OF THE INTERNAL ANGLES OF TRIANGLE ALPHA, BETA, GAMA
CALCULATION OF THE TRIANGLE SURFACE S
PRESENTIG THE RESULTS:
A, B, C;
ALPHA, BETA, GAMA;
S
END

Figure 3-2: Flowchart of the program "triangle"

3.2.2.1 Functional testing (Black-box testing, Data-driven testing)

In this testing mode, the tester treats the program as a black box. Test data are prepared solely on the basis of specifications. One approach to detect all (!) errors in the program is to perform the so-called "exhaustive testing of inputs", or totally functional testing. This is the testing in which all possible combinations of values of input parameters are used as test cases, and not just their reasonable values. In the case of the“triangle” program for the testing of only one function – e.g. the function of detection of equilateral triangle, if the lengths of the sides were unsigned 16-bit “int” type, the application of all possible combinations of sides in lengths between 0 and 65535, means 2.81474976710656 10^{14} possible test cases! If the execution of each test case took 1 µs, this would mean 8.9 years!!
Although this is a huge number of test cases, it does not cover by far all the possible values of input parameters, for example, when as input parameters (sides) one entered -3, -3, -3 (negative numbers) or A, K, Z (characters). Last but not least, the operation of certain programs is conditioned by some historical data (for example, in the case of a program for booking airline tickets – booking a flight for a passenger on the waiting list, if someone else cancels the flight on the same aircraft). This means that, in order to meet the criterion of testing the program with all possible combinations of input parameters, we need to add all possible sequences of all possible combinations of input parameters. Obviously, exhaustive testing of inputs is not feasible in practice. The result of this fact is that the tester can never be sure that the program is completely free of defects. Knowing that a good software testing manager needs to consider the rules of economy and raise the objective of testing as maximizing the added value (number of errors discovered) with definitive investment (the number of test cases). In practice this means that, even by a quick review of the program code, we may find that the program will function properly, both for the value of the input parameters 2,2,2, and the value of 3,3,3, and can optimise the test cases taking this into consideration.
It is very much the same in the case of calibration of a measuring instrument – measurements are never performed at all points of the measuring instrument's measuring range. Instead, several points within the measuring range are selected (based on technical specification, knowledge of the measuring instrument's design, experience, …). If necessary, values between those points are estimated by interpolation, extrapolation or a curve fitting algorithm.

3.2.2.2 Structural testing (White-box, Glass-box, Logic-driven testing)

The object of testing using this strategy is the internal structure of the program. The tester defines the test data on the basis of a review of the program source code. For the inexperienced tester the logical conclusion will be that the test data have to be selected so that each software statement is executed at least once. This means that the program is fully tested when all the logical paths within the logical structure of the program are executed at least once. Such approach is called the "comprehensive testing path" (path exhaustive testing).
Similarly as in the case of "exhaustive testing of inputs”, the number of combinations of a unique logical path is extremely high, which will result in an extremely long time needed to complete the

testing. Such aproach is not feasible in practice. Another shortcoming of this approach is that the implementation of all routes in the program does not mean that the program is fully tested according to the specifications. A simple illustration of this statement is, e.g,. a case, where the programmer made the following error: the condition statement was written as:

"if (a-b) < epsilon" rather than "if |a-b| < epsilon"

The discovering of this error depends on the values of variables a and b, and the error will not necessarily be detected, even though all program paths have been executed.

Although the method of comprehensive testing of the inputs has certain advantages over the method of comprehensive path testing, neither of them is useful in practice. An acceptable and feasible testing strategy can only be a combination of both.

3.2.3 Rules in software testing

The definition of a test case must necessarily include the expected result (the reference value). The tester should know exactly what can be expected as a result of the test case, because in this way he might find errors much more easily.

The programmer or organisation that has developed a program has to avoid testing their own programs. On the one hand, the reasons for this are of psychological nature – no one likes to admit that they have made a mistake, and not many people are willing to look for their own mistakes. On the other hand, the nature of the problem is completely logical – if the programmer has erroneously understood the requirements, and consequently, the program does not do what it should, the same misunderstanding will happen in the testing phase.

Test cases must include both the irregular and unexpected, and the regular and expected values of the input parameters.

During testing it is necessary to make sure that the program does not do what it is not expected to do, and that the program does what it is intended to do.

The probability of existence of some additional errors in the code segment is higher when any previous fault has already been discoveredin this segment.

3.3 Software testing techniques

There are two basic types of software testing techniques:

- Static techniques, which consider reviewing and not executing the program code (the testers read-examine the code), and
- Dynamic techniques, which consider executing the program code.

3.3.1 Static testing

In the early periods of software technology it was considered that, since the programs had been written for computers, the only way to test them was by execution on the computer. Hovewer, some errors became obvious much sooner after reading the source code than after long-lasting execution of the program with implemented numerous test cases. In the seventies, another opinion emerged, i.e., that an effective way of testing and assessment of errors was to read the programs as well. These techniques are particularly useful in the early stages of testing – during and after the coding stage and before the phase of testing by execution of the program.

Although the use of these techniques in testing is informal and subjective, they are largely contributing to productivity and reliability. Firstly, fault detection in the early stages of testing results in lower costs of fault elimination, and in inserting the minimum number of new errors (which is very important). The second factor is primarily psychological – at this stage, pressures of the management (due to development deadlines and delivering the program to the customer, or placing the program on the market) are lesser. The work of the programmers, who have to remove the discovered error as soon as possible, is much more relaxed (and during the elimination of errors they introduce fewer new errors) than during the testing by executing the program on the computer.

The most frequently used static test techniques are inspection, walkthrough and review. The inspection and walkthrough techniques are used in basic, non-automated testing of software. Their essential feature is reading or visual inspection of the source code. A group of testers read the source code of the program and discus the solutions. The programmer explains how and why something has been done, and he often discovers the error by himself during the explanation.

Another similar method is called "desk checking" (an informal review of the code, which is performed by the programmer before start of the testing). Inspection and walkthrough yield significantly better results, because "independent" persons participate in the review of the code. Experience has shown that using inspection and walkthrough it is possible to discover 30–70% of the logical design and coding errors. By inspection and walkthrough it is of course not possible to detect errors that result from incorrect specifications of the program. Inspection and walkthrough are especially useful in testing changes to the program, which, according to experience, involves more errors than writing a new program.

Certain errors can not be detected using the static test methods. The static and dynamic test methods are complementary – each group is appropriate for detecting a specific set of errors. For the proper software testing we have to implement both.

Table 10-10 (Section 10.4) provides a list of the common and most frequent programming errors.

3.3.2 Dynamic testing

Dynamic testing is the testing of the software by executing it. The core problem of dynamic software testing is the selection/preparation of test cases.

It has already been explained herein how much time would be required to complete the testing by the method of exhaustive input testing, and by the method of exhaustive path testing. Random selection of input parameters is not profitable; it is very unlikely that the errors in the program are randomly distributed. It is therefore necessary to define the criteria for the selection of a minimum, but sufficient number of test cases to discover enough programming faults.

A test case is an entity, which is composed of input data, the expected result and the conditions under which it should be used. Considering the fact that the complete program testing is not achievable, a logical conclusion arises that some errors in the program can never be discovered. The aim of software testing is to minimize the number of remaining errors. If we once again recall the definition of a good test case, saying that a good test case is not the one that shows that the program is functioning properly (waste of time and money). A good (successful) test case is the one that causes the error. Therefore, the proper selection of test cases – which will provoke maximum number of errors – is extremely important. The question is which subset from the multitude of all possible test cases has the highest probability of detecting the maximum number of errors?

Neither exhaustive input testing nor exhaustive path testing are the strategies used in practice. Practice has shown that an appropriate combination of elements of these two strategies is a useful testing strategy. It is recommendable to start the testing process by functional testing, and then supplement the testing process by checking the structure of the program under test.

There are several approaches to testing the logic of the program; the most frequently used are logic coverage, code coverage, branch coverage, decision coverage, and decision condition coverage.

The most effective methods for the preparation of test cases are the method of equivalence partitioning (division of the program input domain into intervals of similar treatment of input parameters), and the method of boundary value analysis (focus on values where decision statements switch control to different segments of the code).

3.3.3 Final approach to software testing

The methods presented in the previous chapter represent the iron repertoire of practical implementation of software testing theory. There are several other, probably more enhanced methods, which, however, will not be explained here.

The testing of a concrete software product in practice requires sensible use of all the above-mentioned techniques and strategies [96]. Each of them contributes its share to building successful test cases.

3.3.4 Levels of testing

Levels of testing are strongly connected with the phases in the software lifecycle. The most important ones are comprised in Table 3-1.

Level of testing	**Description**
Module testing (Unit; module, component testing)	Testing of individual program modules, subroutines or procedures. A module is the smallest unit (component) of a software product that can be independently compiled, edited, embedded to a library, or tested. In the lifecycle of a software component, module testing is placed between the phases of module coding and integration of software product. Since the volume of a program code is small (e.g. up to 500 program statements), locating and eliminating errors is easier. The documentation necessary for preparing the test cases includes functional specification of the module and its source code. As a first step it is recommended to test the logic of the program using one of the structural test methods (code coverage, decision coverage, condition coverage). Next step involves functional testing, for which the test cases may be defined according to the functional specification of the module. The general rule is that the modules performing the critical or input/output functions are tested as soon as possible and are preferably replaced by the substitute modules.
Function testing	Testing of the functions is an attempt to detect disagreements between the specification and the external specification. Except for very small programs, the appropriate approach is black box. Test cases are defined on the basis of analysis of the external specification.
System testing	Testing of the system involves comparing the system or program to its original objectives, which are defined in the requirements specification (system specification). Therefore, the project documents need to contain measurable objectives. The problem arises because the objectives do not directly derive test cases. The best way for preparation of test cases is therefore reading the specifications from the user's manual. This enables simultaneous testing of the user's manual. The most important elements of system testing are listed in Table 10-11 (Section 10)
Acceptance testing	Acceptance testing involves the process of checking the program against the initial specification of requirements (or terms of the contract) and the objective needs of the user. It is normally carried out by the customer or end user trying to show that the program does not meet the specifications.

Table 3-1: An overview of the software testing levels

Testing a large software product means detecting and eliminating thousands of errors, creating tens of thousands of test cases, involving large numbers of people months ... It is inevitable to manage this process well. The starting point is a good plan followed by continuous monitoring. Elements of a good plan are listed in Table 10-12 (Section 10.6).

3.3.5 Test completion criteria

This is not a trivial issue at all. The criteria may be different – after certain time of testing, after detection of a certain number of errors, when there are no errors detected for certain time of testing [95, 103]. If the activities during the time provided for testing are not clearly defined, one can spend the time doing improper things. The criterion of test cases not discovering any errors is very bad in the case of e.g. applyings improper test cases. A little better criterion also contains a definition of the test methods used, e.g.: “The program is sufficiently tested when the test cases, designed by the method of covering multi-conditional decisions and analysis of boundary values, do not reveal new errors.”

Definition of the target number of discovered errors is not a good criterion, either. There are many undeterminable quantities, e.g.: total number of errors in program, number of errors to be detected and removed, and acceptable number of remaining errors. All the errors are not equally dangerous.

The number of errors may be estimated on the basis of experience. The literature [28] says that the average number of errors for programs at the end of the coding phase is 4–8 programming errors per 100 program statements.

Taking a program with 10,000 software statements as an example, we can estimate that there are 5 remaining errors per 100 software statements. Theassumed objective is the detection of 98% of coding errors and 95% of design errors. The estimated total number of errors is 500, of which 200 are assumed to be coding errors and 300 design errors. Furthermore, the distribution of the discovered errors by testing phases is presumed – the example results in a Table 3-2 with goals and criteria for the end of each phase of testing.

Testing phase	**Coding errors**	**Design errors**	**Total errors**
Module testing	65% (130)	0% (0)	26% (130)
Function testing	30% (60)	60% (180)	48% (240)
System testing	3% (6)	35% (105)	22.2% (111)
Total	98% (196)	95% (285)	96.2% (481)

Table 3-2: Estimation and distribution of errors per testing phase

Criteria for the completion of testing can be combined – e.g. time of testing, number of discovered errors per unit of time, and total number of discovered errors.

3.3.6 Tools for automation of the software testing and other methods

It is possible to automate software testing to a certain extent. There are several commercially available tools. The majority of tools are internal – built for purpose in testing institutions or departments. Such tools are not generally applicable or available, either because they are specially designed for particular applications, or the authors do not want to publish them (because they think that thanks to these tools they are better than the competition). The automation of a generation of test cases is in principle a difficult task. Such automation tools are, for example:

- module driver tools (for testing isolated modules),
- static flow analysis tools (for testing the use of variables, segments of the code, interfaces between modules),
- coverage analysers (finding the “dead code”),
- environmental simulators,
- hidden path analysers, and
- code instrumentation tools.

3.4 Testing the security of program and data [49]

Although the security was originally treated within the scope of software characteristic "functionality" [23], it has recently evolved as a stand-alone, very important issue that requires special consideration. The testing and evaluation of software security within the scope of information technology security is carried out according to special security criteria laid down in international regulations, such as:

- ITSEC: Information Technology Security Evaluation Criteria[1],
- ISO/IEC 15408-1, 2, 3: 2005-2008 (CC): Common Criteria for Information Technology Security Evaluation [19].

Conformity tests are mainly carried out for the following quality characteristics:

- Availability: data and services must be available to authorised users at any time;
- Confidentiality: information shall be available to authorised users only;
- Integrity: data and programs must be protected from unintended or unauthorised modifications (including complete loss);
- Authenticity: programs must clearly identify the communication partner (user, process) of protected transactions.

In particular the existence and efficiency of security measures is tested. Such measures can be:

- controls of access to programs, password systems,
- restrictions of access to certain data storage areas or documents,
- role concepts, graded granting of rights,
- measures for anti-virus protection, firewalls,
- plausibility checks for all input data,
- database integrity rules,
- archiving and backup measures, and
- disaster recovery.

An analysis conducted during the MID-Software project checked whether the Common criteria approach fitted the needs of conformity assessment of the software in legal metrology. It has been ascertained that legal metrology applications are not yet ready for such a mature approach [13], neither regarding the risk of application nor in technological realisation of applications and knowledge of legal metrology experts. Practice has shown that even very simple methods, e.g. for checking the integrity of programs and measurement data (CRC-, hash functions), and elements of the PKI infrastructure, almost completely fulfill the security requirements for state-of-the-art legal metrology applications. However, this area has to be carefully considered in the future.

[1] ITSEC (June 1991). *Information Technology Security Evaluation Criteria (ITSEC): Preliminary Harmonised Criteria*. Document COM(90) 314, Version 1.2. Commission of the European Communities.

3.5 Testing the software process [49]

It is necessary for a high-quality software product that its overall development process and all its sub-processes are of high quality, too. This is achieved through assessing the overall process and its sub-processes, and improving them on the basis of the assessment results [94]. The assessment of processes and their improvement are cyclical elements.

Validation of the software process comes into account for conformity assessment module H1 [14]. Formal or informal assessments of software development processes may be applied in the form of audits at the software producer's site. On the one hand, formal assessment systems are offered by the International Standards series ISO/IEC 15504 (SPICE) [20], or the assessment models CMMI (Capability Maturity Model Integration). On the other hand, in informal assessments, made-to-measure lists of questions based on relevant process standards (such as ISO/IEC 12207) [18] can be compiled and applied. If the answers to the audit questions are positive, and if the overall assessment of the development process is positive, too, this can be regarded as an important indication that the producer's software is in compliance with the latest state of the art. All in all, it can be assumed that in the case of an overall positive judgement, good testability and maintainability of the software product is ensured. An example of such a collection of items to be checked may be:

a. Software testing

- Are adequate software testing procedures used in the project (test planning, test case determination, test data generation, performance of tests, test analysis, test documentation)?
- Are the methods used documented?
- Who carries out the tests? Is there a separate group for this?
- Are reviews of the results of the different stages carried out at regular intervals (e.g. review of the requirements specification, design documents, test schedules)?
- Who draws up the test schedules and by what means?
- Who carries out the functional test?
- Are there test records and summarising test reports?
- Is there a procedure to record and archive the test results?
- By what means is it assured that all user requirements and product functions have been checked?
- Who authorises test schedules, test records, test reports?
- Are confidence-building internal audits of the software development processes carried out?
- Can the producer guarantee access to the test environment (testing tools, test data) and to the test documentation?

b. Software documentation

- Are there any company-specific standards or guidelines for the preparation and maintenance of the software documentation?
- Is the documentation compiled parallel to the project?
- Is the documentation subject to regular reviews?

- What is the scope of the documentation supplied?
- For how long is the documentation valid?
- Is the documentation subject to version management?
- Can the producer guarantee access to the software development documentation?
- Can the producer guarantee access to the source code (if necessary)?

c. Error messages and change management

- Does a formal problem message procedure with feedback to the customer or to the test team exist?
- How does the producer react to error messages and suggestions for improvement, and how does he deal with them?
- In what way are the customers or the test team informed about how the matter is handled?
- Are error statistics (error types, frequencies) compiled?
- Who initiates software modifications?
- Who approves software modifications?
- Are there documented methods for change and version management?
- Do the procedures of the change management also apply – apart from programs – to the software documentation and other documents such as test schedules, test reports or design schedules?
- Are new software and document versions systematically identified?
- How are the customers informed about new versions?
- Is a configuration management tool used?
- Are there any provisions as to which tests have to be repeated in case a version is modified, and in which way they have to be carried out?

If necessary, these audit questions have to be extended or refined to allow producer- or domain-specific checklists to be derived.

Finally, the quality of the software development processes can be improved by the following means:

- The results of the audits having been carried out at the producer's are directly put into practice (this has a direct impact on the processes and, consequently, on the products);
- The use of auxiliary means in software development: standards, working instructions, checklists, process models, guidelines (e.g. for design, programming, documentation, testing);
- The use of validated software tools (compiler, configuration management);
- The use of validated requirements catalogues;
- The re-use of high-quality software or software components.

The software process testing approach is applicable to the conformity assessment module H1 for legal measuring instruments (Section 1.4).

3.6 Testing the Internet applications [28]

Taking into consideration the rapidly growing use of the Internet for metrological applications (as explained in Section 2.1), it is obvious that this area needs special consideration. Although the testing of such applications does not differ significantly from classical software testing approaches explained in the previous sections, some specific and systematic approaches arise from the structure of the applications.

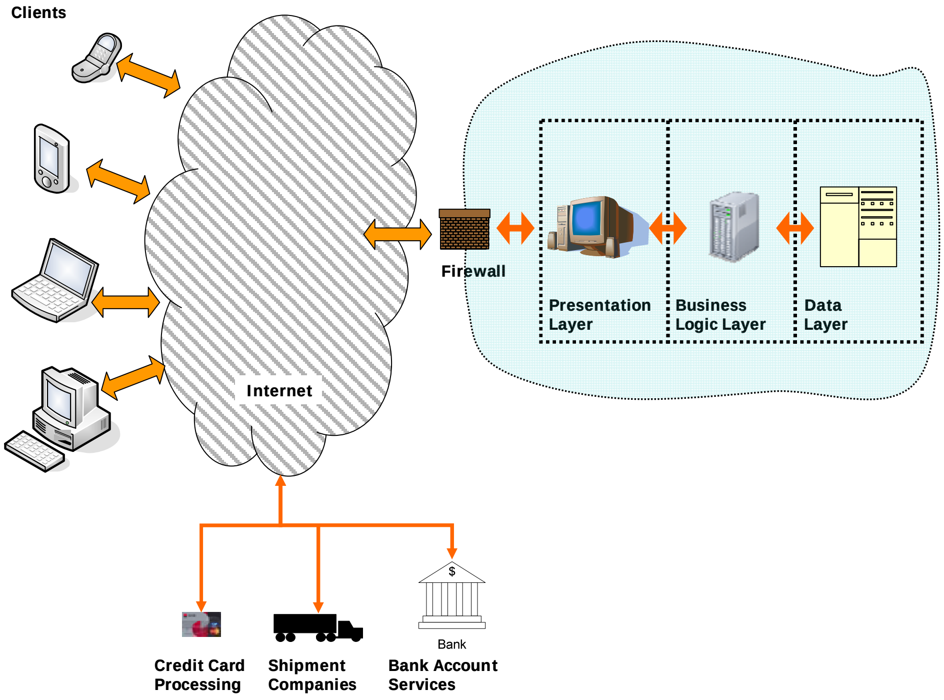


Figure 3-3: Internet application - a typical structure [28]

Internet applications are the usual client-server applications with the Web browser as a client who receives service from the Web or application server. However, clients have access to the applications not only from different Web browsers running on personal computers, but very often from mobile phones. In addition, modern refrigerators, cell phones, or cars, already have or are expected to have an embedded Internet connection functionality.

The application has three essential building blocks (tiers, layers). Each building block may be treated as s black-box with well defined interfaces.

The customer’s point of contact with the application is the Web server, often called the Presentation Tier or Layer, since it gives the first presentation of the application to the customer.

The Business Layer contains software that models the business process, running on the application server. The following functionality is realised here:

- transaction processing,
- user authentication,
- data validation, and
- application logging.

The third, Data Layer, takes care of storing and retrieving data; it is typically a relational database management system. Depending on the business size, it may consist of several database servers and additional functionality, such as an authentication server.

There are many possible failure points in an Internet-based application. The following list is a reminder on the issues one has to keep in mind when testing Internet-based applications:

- Large and diverse user base. The users of an Internet application possess different skill sets, employ a variety of browsers, and use different operating systems or devices. They obtain access the the application Website using a wide range of connection speeds.
- Business environment requires a repertoire of functions such as calculating taxes, determining shipping costs, completing financial transactions, and tracking customer profiles.
- Various regional settings. Users may reside in other countries, in which case internationalization issues will arise, such as language translation, time zone considerations, and currency conversion.
- Testing environments. Proper testing of the application requires duplication of the production environment. This means that the tester should use Web servers, application servers, and database servers that are identical to the production equipment. For most accurate testing results, the network infrastructure will have to be duplicated as well. This includes routers, switches, and firewalls.
- Security – protection from hackers (provoking unwanted denial-of-service (DoS) attacks or ripping off customers' credit card information).

Presentation Layer	**Business Layer**	**Data Layer**
Ensure fonts are the same across browsers.	Check for proper calculation of sales tax and shipping charges.	Ensure database operations meet performance goals.
Check to make sure all links point to valid files or Websites.	Ensure documented performance rates are met for response times and throughput rates.	Verify data are stored correctly and accurately.
Check graphics to ensure they are the correct resolution and size.	Verify that transactions complete properly.	Verify that you can recover using current backups.
Spell-check each page.	Ensure failed transactions roll back correctly.	Test failover or redundancy operations.
Allow a copy editor to check grammar and style.	Ensure data are collected correctly.	
Check cursor positioning when page loads to ensure it is in the correct text box.		
Check to ensure default button is selected when the page loads.		

Table 3-3: Testing Internet application – basic hints [28]

It is evident that configuring a testing environment provides one of the most challenging aspects of E-commerce development. Testing applications that process financial transactions requires most effort and expenses. All the components used for the application to produce valid test results have to be replicated, both hardware and software.

Another significant testing challenge is testing browser compatibility. There are several different browsers on the market today, and each behaves differently. Although standards exist for browser operation, most vendors enhance their browsers to try and attract a loyal user base. Unfortunately, this causes the browsers to operate in a non-standard way. Although many challenges exist when testing Internet-based applications, testing efforts should be focused on specific areas. Basic hints for testing are presented in Table 3-3.

In the Internet environment, it is critical to keep the Website available for customer use. This requires preparation and implementation of maintenance guidelines for all the supporting applications and servers. Items such as the Web server and RDBMS[2] require a high level of management. Logs, system resources, and backups require special attention.

Last but not least, network connectivity provides another area on which to focus the testing efforts. At some point, network connectivity can go down. The source of the failure might be the Internet itself, the service provider, or the company's internal network. Therefore, it is necessary to prepare contingency plans for the application and infrastructure to respond gracefully when an outage occurs.

As it is critical to successful testing of standard applications, a specification document is needed to describe the expected functionality and performance of the Website. Without this document, one cannot design the appropriate tests.

Both the components developed internally and those purchased from a third party should be tested. For testing the in-house developed components the tactics presented in earlier chapters applies.

For purchased components it is necessary to develop a series of system tests to validate that the items perform correctly independently of the application. In this case it is necessary to apply the purchaser approach and perform the acceptance testing approach. Ideally, this test should be completed independently of the application testing. Including a non-functional third-party component in the architecture of the final system makes it difficult to interpret the test results and identify the source of errors. Generally, black-box approaches apply to third-party components, because you rarely have access to the component internals.

Testing Internet-based applications is best tackled using a "divide-and-conquer" approach. Fortunately, the architecture of Internet applications allows the identification of discrete areas to target testing. Figure 3-3 illustrates the basic architecture of Internet applications, while Table 3-4 lists the items to be tested in each layer. The list is not complete, but provides a starting point for developing the testing criteria.

[2] RDBMS: Relational Database Management System

Test Area	Comments
Usability /human factors	Review overall look and feel. Fonts, colours, and graphics play a major role in the application aesthetics.
Performance	Check for fast-loading pages. Check for quick transactions. Poor performance of ten creates a bad impression.
Business rules	Check for accurate representation of business process. Consider business environment for target user groups.
Transaction accuracy	Ensure transactions complete accurately. Ensure cancelled transactions roll back correctly.
Data validity and integrity	Check for valid formats of phone number, e-mail addresses, and currency amounts. Ensure proper character sets.
System reliability	Test the failover capabilities of your Web, application, and database servers. Maximize MTBF and minimize MTTR.
Network architecture	Test connectivity redundancy. Test application behaviour during network outages.

Table 3-4: Items to test in each layer

Testing the Presentation Layer consists of finding errors in the GUI, or front end, of the application. This important layer provides the first contact of the costumer to the site, so detecting and correcting errors in this layer are critica1 to presenting a quality, robust Website. If the customers encounter errors in this layer, they may not return. They may conclude that if the company creates Web pages with misspelled words, it cannot be trusted to successfully execute a credit card transaction. However, just as testing of an Internet application can be segmented into discrete entities, the same can be done when testing the Presentation Layer. The following identifies the three major areas of Presentation Layer testing:

- Content testing; overall aesthetics, fonts, colour, spelling, content accuracy, default values;
- Website architecture,– broken links or graphics;
- User environment–web browser versions and operating system configuration.

The Business Layer testing focuses on finding errors in the business logic of the Internet application. Testing this layer is very similar to testing stand-alone applications – both white- and black-box techniques may be applied. Test plans and procedures are needed that detect errors in the application's performance requirements, data acquisition, and transaction processing.

White-box approaches should be applied for components developed in-house because of the availability of the source code. However, black-box testing techniques are the primary testing approach for this layer. The starting point will be developing test drivers to unit-test the individual components. Next, a system test should be performed to determine whether all the components work together correctly. When conducting a system test for this layer, you need to mimic the steps a user performs when purchasing a product or service.

The technologies used to build the business logic dictate the building and conducting of the tests. There are numerous technologies and techniques to build this layer, which make it impossible to suggest a universal test method.

Regardless of the approach, there exist certain characteristics of the application that should always be tested. These areas include the following:

- Performance. Test to see whether the application meets documented performance specifications (generally specified in terms of response times and throughput rates).
- Data validity. Test to detect errors in data collected from customers.
- Transactions. Test to uncover errors in transaction processing; this may include items such as credit card processing, e-mailing verifications, and calculating sales tax.

Once the site is up and running, the collected data become very valuable (e.g. credit card numbers, payment information, and user profiles). Losing this information could have disastrous consequences for the business. Testing of the Data Layer consists primarily of testing the database management system that the application uses to store and retrieve information. Smaller sites may store data in text files, while more complex sites use full-featured enterprise-level databases.

One of the biggest challenges associated with testing this layer is duplicating the production environment. You must use equivalent hardware platforms and software versions to conduct valid tests. In addition, once you obtain the resources, both financial and labour, a methodology for keeping the production and test environments synchronised should be implemented.

Certain specific errors need to be eliminated during the testing of the Data Layer. These include the following:

- Response time. Quantifying the completion times for Data Manipulation Language (DML), queries (SELECTs), and transactions. Response-time testing in this layer does not include timing page loads, but focuses on identifying database operations that do not meet the performance objectives. When testing the Data-Layer response time, it is necessary to ensure that individual database operations occur quickly, so as not to bottleneck other operations.
- Data integrity. Verifying that the data are stored correctly and accurately.
- Fault tolerance and recoverability. Maximize the MTBF[3] and minimize the MTTR[4].

[3] Mean Time Between Failures

[4] Mean Time To Recover

4 Quality Requirements for Metrological Software [37]

This Section presents the results of a survey regarding the available guidance documents on quality and validation of the metrological software, which was performed in the course of preparation of the WELMEC Guide 7.2. The aim of the analysis was to gather information about existing approaches to validation of metrological software that could be, by segments, applicable to the task – preparation of a guidance document for validation of the software of measuring instruments covered by the MID. Another important aim was to check that the newly developed guidance was not in contradiction with the existing approaches.

4.1 Introduction

The quality characteristics of metrological software are required by the national or regional legislation, e.g. in legal metrology, or for safety critical applications, or by various standards such as software product quality standards (ISO/IEC 9126) [23] or laboratory competence standards (ISO/IEC 17025) [21].

The parties involved were interested in getting a clear guidance for software quality requirements and validation methods. Several guidance documents have already been developed. In these documents, different software quality issues and software lifecycle phases are addressed to different extents, as well as the outcome of risk evaluation for software malfunction or fraud.

With the intention to support the quality evaluation of such metrological software, several guidance documents have already been generated, such as the WELMEC Guides (2.3 [39], 2.5 [40], 7.2 [41]), the OIML D-31 [30] Guide, the EUROLAB TR 2/2006 [11], the NORDTEST Software Validation Guide [29], the Canadian “Terms and Conditions for the Approval of Metrological Software”[27], and the FDA Software Validation [38] and Electronic Signature [12] Guides.

4.2 Definition of the equipment under test

The validation guidance documents address a wide variety of metrology software and other IT applications, ranging from applications directly related to the measurement process, to administrative procedures and management documents. The software subject to validation may be commercial off-the-shelf (COTS), modified off-the-shelf (MOTS), or custom-developed (designed for a particular application). Although it can be found in the literature that commercial off-the-shelf software used within its designated application range may be considered to be sufficiently validated [21], this is not always true [98].

Most guidance documents deal with the final software as a part of the measuring instrument and its validation, i.e. the “confirmation by examination and provision of objective evidence that the requirements for a specific intended use are fulfilled”. In some guides, however, other software lifecycle phases are addressed as well. Consequently, these guides deal with verification of particular software lifecycle phases, i.e. the “provision of objective evidence that the design outputs of a particular phase of the software development life cycle meet all of the specified requirements for that phase”. Hence, one may say that two major groups of guidance document users are addressed. The first group is interested in validation only. The second group deals with the development of metrological software applications, and needs guidance for verification of particular software lifecycle phases.

It is very important to consider that the term 'verification' has different meanings in the legal metrology and other communities dealing with testing and validation. In legal metrology, the term verification means a “procedure (other than type approval) … that ascertains and confirms that the measuring instrument complies with the statutory requirements [16]” . This understanding is close to the definition of validation.

A good definition of a subject of validation can be found in the FDA[1] Guide [38]:

- software used as a component, part, or accessory of a medical device,
- software that is itself a medical device (e.g. blood establishment software),
- software used in the production of a device (e.g. programmable logic controllers in manufacturing equipment), and
- software used in implementation of the device manufacturer's quality system (e.g. software that records and maintains the device history record).

4.3 Target audience

A good definition of possible categories of users of guidance documents can be taken from the FDA SW Validation Guide [38]. To apply this approach to the field of measuring instruments, it will be enough to replace the term “medical device” by the term “measuring instrument”, and FDA by “inspection body”:

- persons subject to the medical device Quality System regulation,
- persons responsible for the design, development, or production of medical device software,
- persons responsible for the design, development, production, or procurement of automated tools used for the design, development, or manufacture of medical devices or software tools used to implement the quality system itself,

[1] U. S. Food and Drug Administration, http://www.fda.gov/

- FDA Investigators,
- FDA Compliance Officers, and
- FDA Scientific Reviewers.

4.4 Sources of requirements

The requirements for the quality of metrological software may be regarded from several expert points of view. These points of view are:

- software technology standards,
- legislation,
- safety-related standards,
- laboratory competence standards, and
- security-related standards.

The basic quality characteristics of software products are defined in the International Standard ISO/IEC 9126:2001 – Software engineering – Product quality [23]. These quality characteristics are functionality, reliability, usability, efficiency, maintainability and portability.

The requirements that have their origin in legislation, safety-related standards and laboratory competence standards, always address functionality, reliability, and usability. Maintainability is rarely addressed.

Software-related requirements in important industrial areas have their roots in risk assessments of the consequences of software malfunctions [100, 101]. Therefore, some key industrial areas have already developed mature requirements, often with respect to software safety and reliability [22]. These areas are, for instance:

a) General:

- EN 61508:2002/2003: Functional safety of electrical/electronic/programmable electronic safety-related systems, Parts 1–7, 2002–2003

b) Transportation industry:

- EN 50128:1997, Railway Applications: Software for Railway Control and Protection Systems;

- Development Guidelines for Vehicle-Based Software, The Motor Industry Software Reliability Association[2] (MISRA™), November 1994;

c) Aerospace industry:

- RTCA[3]/DO-178B: Software Considerations in Airborne Systems and Equipment Certification;
- ECSS[4]-Q-80B: Software Product Assurance;

c) Nuclear power industry:

- IEC[5] 60880:1986–09, Software for Computer in Safety Systems of Nuclear Power Stations;

d) Medical devices:

- Medical Device Quality System Regulation (FDA, CHR: Title 21, Food and Drugs, Subchapter H, Part 820: Quality System Regulation, 1997)

E-banking is another example of an area which would seriously suffer from software malfunctions or frauds. This and other areas benefit from the EU Directive on Community framework for electronic signatures (1999/93/EC), or the International Standard ISO/IEC 15408, Information technology – Security techniques – Evaluation criteria for IT security (common criteria).

Generally speaking, the risks of failure in metrological software applications are not so high, compared to other areas. However, they are certainly not negligible, as they vary from cheating customers when measuring instruments are used for direct sale, to risking serious health hazard caused by malfunction of a biomedical measuring instrument.

4.5 Requirements for software in legal metrology

Examples of legislative documents concerning metrological software are the European Directive on non-automatic weighing instruments (NAWI) [3] and the Measuring Instruments Directive (MID) [4], which covers 10 groups of measuring instruments. The quality requirements for metrological software are very rarely stated as explicit requirements. In most cases, they are hidden in the context of the Directives.

[2] http://www.misra.org.uk/

[3] Radio Technical Commission for Aeronautics (http://ww.rtca.org)

[4] European Cooperation for Space Standardisation (http://www.ecss.nl/)

[5] International Electrotechnical Commission (http://www.iec.ch/)

4.5.1 Directive 90/384/EEC on non-automatic weighing instruments (NAWIs)

The requirements for metrological software in the NAWI Directive are included in the essential requirements for the protection against modifications, manipulation or fraudulent use of non-automatic weighing instruments:

- "Design and construction of the instruments shall be such that the instruments will preserve their metrological qualities when properly used and installed, and when used in an environment for which they are intended..."
- "The instruments shall have no characteristics likely to facilitate fraudulent use, whereas possibilities for unintentional misuse shall be minimal. Components that may not be dismantled or adjusted by the user shall be secured against such actions."
- "The applicant shall keep the notified body that has issued the EC type-approval certificate informed of any modification to the approved type."

Obviously, for validation purposes, these requirements have to be interpreted in order to be applicable to software and legal metrology validation and verification. The guidance documents follow the development of technology. At the beginning, only the software embedded in a measuring instrument was important. With the expansion of personal computers, point-of-sale devices became important as well. As a result, two guidance documents were produced concerning the software of NAWIs:

- WELMEC 2.3: Guide for examining software [39],
- WELMEC 2.5: Guide for modular approach and testing of PCs and other digital peripheral devices [40].

4.5.2 Directive 2004/22/EC on Measuring Instruments (MID)

The MID addresses ten groups of measuring instruments that are mostly under legal control in European countries. These instruments are water meters, gas meters and volume conversion devices, active electric energy meters, heat meters, measuring systems for continuous and dynamic measurement of quantities of liquids other then water, automatic weighing instruments, taximeters, material measures, dimensional measuring instruments, and exhaust gas analysers.

Compared to the NAWI Directive, the MID has more specific requirements for the software of measuring instruments. Examples of the requirements in MID, Annex I, are listed in Table 4-1 below.

Clause	Requirement
7.	Suitability
7.6.	A measuring instrument shall be designed so as to allow the control of the measuring tasks after the instrument has been placed on the market and put into use. If necessary, special equipment or software for this control shall be part of the instrument. The test procedure shall be described in the operation manual. When a measuring instrument has associated software which provides other functions besides the measuring function, the software that is critical for the metrological characteristics shall be identifiable and shall not be inadmissibly influenced by the associated software.
8.	Protection against corruption
8.1.	The metrological characteristics of a measuring instrument shall not be influenced in any inadmissible way by the connection to it of another device, by any feature of the connected device itself or by any remote device that communicates with the measuring instrument.
8.3.	Software that is critical for metrological characteristics shall be identified as such and shall be secured. Software identification shall be easily provided by the measuring instrument. Evidence of an intervention shall be available for a reasonable period of time.
8.4.	Measurement data, software that is critical for measurement characteristics and metrologically important parameters stored or transmitted shall be adequately protected against accidental or intentional corruption.

Table 4-1: Software requirements in MID [4]

4.6 Requirements in the laboratory competence standards

Examples of such standards are [5]:

- ISO ISO/IEC 17025:2005 General requirements for the competence of testing and calibration laboratories;
- ISO 15189:2007 Medical laboratories – Particular requirements for quality and competence;
- ISO/TS 16949:2009 Quality management systems – Particular requirements for the application of ISO 9001:2000 for automotive production and relevant service part organizations; and
- ISO 15161:2001 Guidelines on the application of ISO 9001:2000 for the food and drink industry.

4.6.1 Software-related requirements in ISO/IEC 17025

Table 4-2 below lists the most explicit software/IT-related requirements in ISO/IEC 17025 [21]

ISO/IEC 17025 Clause	Requirement
5.4.7.2a	When computers or automated equipment are used for the acquisition, processing, recording, reporting, storage or retrieval of test or calibration data, the laboratory shall ensure that computer software developed by the user is documented in sufficient detail and is suitably validated as being adequate for use.
5.5.2	Equipment and its software used for testing, calibration and sampling shall be capable of achieving the accuracy required and shall comply with specifications relevant to the tests and/or calibrations concerned. Calibration programmes shall be established for key quantities or values of the instruments where these properties have a significant effect on the results. Before being placed into service, equipment (including that used for sampling) shall be calibrated or checked to establish that it meets the laboratory’s specification requirements and complies with the relevant standard specifications. It shall be checked and/or calibrated before use.
5.5.4	Each item of the equipment and its software used for testing and calibration and significant to the result shall, when practicable, be uniquely identified.
5.5.10	When intermediate checks are needed to maintain confidence in the calibration status of the equipment, these checks shall be carried out according to a defined procedure.
5.5.11	Where calibrations give rise to a set of correction factors, the laboratory shall have procedures to ensure that copies (e.g. in computer software) are correctly updated.
5.5.12	Test and calibration equipment, including both hardware and software, shall be safeguarded from adjustments which would invalidate the test and/or calibration results.

Table 4-2: List of the direct software-related requirements of ISO/IEC 17025:2005 [21]

Several clauses of the standard deal with requirements for document control (kept in a laboratory or during transmission). Documents may be, e.g. measuring results or contracts with customers. These requirements address both standard IT system administration practices (e.g. roles and access rights) and document management practices (office procedures). Such clauses are, e.g.: 4.1.5c, 4.3.3.4, 4.3.1, 4.3.2.1, 4.3.2.2, 4.3.3.2, 4.4.1a, 4.6.1, 4.6.2, 4.6.3, 4.12.1.2, 4.12.1.3, 4.12.1.4, 4.12.2.1, 4.12.2.2, 4.12.2.3, 5.4.1, 5.4.7.1, 5.4.7.2b, 5.4.7.2c, 5.5.5, 5.10.1, 5.10.2j and 5.10.7.

4.7 Guidance documents

This section overviews those guidance documents that have been analysed. The intention of all the documents is to refine the functional requirements, so that understandable software and other IT requirements and guidance for their validation are available.

4.7.1 WELMEC 2.3 Guide for Examining Software (Non-automatic Weighing Instruments) [39]

<table>
<tr><th>Requirement</th><th>Validation Guidance</th></tr>
<tr><td>Protection of the legally relevant software:
The legally relevant software shall be protected against intentional changes by common software tools.</td><td>- Check whether checksum(s) are generated and that they comply with the documentation.
- Verify the protection means by using a text editor.</td></tr>
<tr><td>Software interfaces
Interfaces between the legally relevant software and the software parts not subject to legal control shall be protective.</td><td>- Check whether the functional description is conclusive.
- Check whether all documented functions or data released or exchanged via the protective interface are allowed.
- Check whether the declaration for completeness is given.</td></tr>
<tr><td>Software identification
There must be a software identification, comprising the legally relevant program parts and parameters, which is capable of being confirmed at verification.</td><td>- Check whether the checksum(s) or other signature(s) are generated and may be confirmed at verification, e.g. by an audit trail.</td></tr>
<tr><td colspan="2">The documentation shall describe:
- All legally relevant parts and parameters of the software.
- The functions of these parts.
- The complete set of commands to be exchanged via the protective software interface.
- A written declaration of completeness of the list of the legally relevant functions and parameters and the documented set of commands.
- The securing measures (e.g. checksum, software identification, audit trail).
- The instructions in order to check the legally relevant software at verification.
- A written declaration that the Standard EN 45501:1992/AC 1993 has been adopted.</td></tr>
</table>

Table 4-3: WELMEC 2.3 – requirements and validation guidance [39]

This document is intended as guidance for the validation of software requirements of the Directive 90/384/EEC on non-automatic weighing instruments (NAWIs).Detailed software requirements and guidance for their validation are given in the Table 4-3.

4.7.2 WELMEC 7.2 Software Guide (Measuring Instruments Directive 2004/22/EC) [41]

The design and intended use of this guidance document, which is one of the most important issues of this thesis, are explained in detail in Section 5.

4.7.3 OIML D-31 General Requirements for Software-Controlled Measuring Instruments [30]

The "Organisation Internationale de Métrologie Légale" (OIML, International Organization of Legal Metrology) and its activities have been explained in Section 1.3.2.2.

The international document D.31 consists of an introduction, an explanation of the scope and field of application, a terminology section, a section that explains the use of the document in drafting OIML Recommendations, a software requirements section, a section with type approval guidance, sections intended for verification, assessment of severity (risk) levels, assessment of software processes, and seven annexes [89].

Only the most important and specific issues are explained below.

4.7.3.1 D-31 Requirements

General software requirements address:

- identification of software,
- correctness of algorithms and functions,
- protection of software from accidental or intentional misuse, and
- support of hardware features (diagnostics).

Specific software requirements define:

- specifying and separating relevant parts and specifying interfaces of parts,
- shared indications,
- storage of data, transmission via communication systems,
- compatibility of operating systems and hardware, portability,
- conformity of production-line devices with the approved pattern, and
- maintenance and re-configuration.

4.7.3.2 D-31 Validation guidance

The description of the software-related part of type approval (software validation) begins with the definition of the necessary documentation to be submitted for a type approval. Furthermore, guidance for the selection of validation procedures with respect to the defined requirements is proposed. The following validation methods are proposed:

- analysis of documentation and specification, and validation of the design,
- validation by functional testing of the metrological function,
- validation by functional testing of the software functions,
- dataflow analysis,
- code inspection and walkthrough, and
- software module testing.

The D-SW also includes a sample software test report and checklists. A procedure proposed for updating legally relevant software of a measuring instrument is of particular interest. The proposed procedure anticipates the conditions for updating software without the need for immediate subsequent verification of the measuring instrument.

4.7.4 Guideline for the use of computers and software in laboratories with reference to ISO 17025 [11],

The intention was to produce a guideline for the use of software and computers in testing and calibration laboratories accredited according to ISO/IEC 17025. The guideline was prepared by a group of authors from laboratories, accreditation bodies and metrology institutes.

The guidance addresses various aspects of IT use in laboratories, including typical office applications, document management software, databases, measurement automation software, firmware, distributed measuring systems, records and documents distribution, security issues, etc.

Two different extents and ways of software quality examination are proposed:

- software validation only, appropriate for users who purchase software,
- software validation and verification, appropriate for users who develop their own software.

There is also an attempt to address the risk of issuing an incorrect test or calibration result.

4.7.5 NORDTEST[6] TR 535: Method of Software Validation [29]

The NORDTEST Software Validation Guide deals with the measurement software only. Although it addresses both purchased and self-developed software products, it provides much more guidance for laboratories that want to develop (and then verify and validate) their own software. Therefore, some software engineering (development) advice for the entire software life cycle is included, such as: specification of requirements and system acceptance test, design process, inspection and testing, installation and system acceptance test, and maintenance.

A validation report template is provided for documenting the validation process.

4.7.6 Canadian “Terms and Conditions for the Approval of Metrological Software” [27]

This guidance has been developed by a workgroup under the direction of the Canadian Federal Department of Industry. It is not intended to apply the guide to relatively simple, built-for-purpose devices, such as electricity or gas meters. The target applications are non-built–for-purpose devices that rely on universal computers as part of the measuring system.

Software is categorised into three categories by function: measurement functions, computational functions, and control functions. The software which performs the measurement, together with the result of measurement value for a certain quantity, belongs to the first category and requires approval. It should be verified prior to being put into service. The requirements do not apply to categories 2 and 3. These categories of software are, however, subject to inspections.

The following list gives a rough survey of the contents:

- Automatic test by the metrological software as to whether the hardware resources allow accurate measurement;
- Automatic test as to whether the code or setup parameters have been inadmissibly changed;
- Integrity of transmitted data;
- Allowed modifications of the approved software;
- Avoidance of loss of measurement values;
- Display of parameter settings;
- Event logger for metrological audit trails;
- Indication of measurement information;
- Provision of a record with all information relevant for the measurement and the price;

[6] http://www.nordicinnovation.net/nordtest.cfm

- Interdiction of installation of metrological software together with software that can adversely affect accuracy; and
- Display of the measurement information free of information not pertinent to the transaction.

The document does not provide guidance for validation of the software.

4.7.7 General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002 [38]

The requirements stem from the FDA Medical device quality system regulation. The Guidance addresses the complete software life cycle and all software components (software in all devices) that influence (during development and manufacturing) the final software-controlled medical device. The Guidance suggests quality planning, including a Risk (Hazard) Management Plan and a Configuration Management Plan. Recommendations for the following lifecycle phases are given: requirements, design, coding, testing by developer, user site testing, maintenance, and software change.

The subject of validation according to this document is already stated in Section 4.2, and the document target audience in Section 4.3.

Several good software engineering practices and activities (planning, verification, testing, traceability, configuration management,…), which provide evidence that software is sufficiently validated, are discussed in this Guidance. Software lifecycle management and risk management activities – and their connection to the development approach, techniques and efforts – are addressed as well. Special attention is paid to the validation of software after modifications, and to the treatment of “off-the-shelf software”.

4.7.8 FDA 21 CFR Part 11, Electronic Records; Electronic Signatures; Final Rule Electronic Submissions; Establishment of Public Docket; Notice [12]

The FDA has developed a separate guidance document that addresses electronic records and electronic signatures. Computer systems used to create, modify, and maintain electronic records, and to manage electronic signatures, are also subject to the validation requirements. Such computer systems must be validated to ensure accuracy, reliability, consistent intended performance, and the ability to detect invalid or altered records.

4.7.9 Standard for Safety-Related Systems, EN 61508 [9]

This is an example of a standard that is not intended for metrology applications, but may nevertheless prove very useful for validation purposes. This comprehensive standard for safety-relevant systems consists of 7 parts. It realises a generic approach and deals not only with software but also with the safety-critical system as a whole, including hardware. It contains detailed guidelines for the complete lifecycle of the safety-relevant hardware and software, including the initial concept, hazard analysis and risk assessment, development of the required safety functions, specification, design and implementation of hardware and software, operation and maintenance, modifications of hardware and software, decommissioning, and disposal.

The development process according to EN 61508 is based on the V-model (similarly as the software development lifecycle V-model): the left branch of the letter „V“ symbolises the development steps in a top-down approach starting with the specification of the safety requirements the system has to fulfil, and ending with the coding step of the software. The right bottom-up branch of the "V" emblematises the test and validation steps where each development step has one or more corresponding test and validation steps. The standard strictly requires the consequent, preferably formal documentation of each step. In particular, in the case of modifications or error corrections, the documentation related to this level and the documentation of the higher levels in the V-model have to be updated – with the benefit that errors at all levels of the safety concept have a better chance to be detected during later lifecycles.

The Standard EN 61508 deals systematically and in-depth with the following items:

- Definition of requirements and description of methods and tools that cover all phases of the system and software lifecycle completely.
- Description of quantitative methods for assessment of the risk of a hazard and of quantitative methods for calculation of the probability of a critical failure.
- For all problems not one requirement or method alone is established but alternatives are given and their appropriateness for the problem is quantified.

Although EN 61508 has been developed in an area adjacent to the metrology, it is discussed here for two reasons: Firstly, the software of measuring instruments and systems is to some extent similar to the software of safety-related devices. Secondly, because of its completeness and comprehensiveness the standard may be regarded as an additional basic guidance for software developers and examiners independently of their branch.

4.8 Comparison – overview of common points and differences between different Guides

A rough overview of the issues covered by the analysed guidance documents is given in Tables 10-4 and 10-5 in Section 10. “Yes” in certain cells means that the item mentioned in the leftmost cell of that row is considered in the guidance document listed in the header row. Since the terminology in the documents is not harmonised, similar things are addressed under different categories, e.g. “software and data protection against unintentional or intentional changes” in one Guide may have the same meaning as “data integrity” in another document.

All documents address identification of software or data record.

Some Guides deal with very specific aspects, such as separation of software into parts under (legal) control and other parts in WELMEC Guides, or monitoring of the temperature of the operating environment of a measuring instrument in the Canadian Guide.

4.9 Conclusions

As a general conclusion, we could say that the different approaches in the requirement and validation guidance documents that have been analysed come from:

- Different risks connected to the application of the software (more strict in medical devices, less strict in legal metrology or testing and calibration, however, with exceptions, i.e., for forensic tests). The intention to deal with risk assessment and its consideration is present in almost all guidance documents. The assessment of risks and the consequences (e.g. the determination of “risk class” or similar categorisation) are, according to the authors’ opinion, subjective rather than comparable in an objective manner. In the case of WELMEC Guide 7.2, for instance, the categorisation of instruments into risk classes is the task of the measuring instruments expert groups.
- The recency of the guidance document.
- Distinction whether the unit under test is a final product and needs (only) to be validated according to the requirements, or the intention of the document is to provide development and verification/testing guidance as well.
- Whether the documents address technical aspects of software products only, or quality system aspects of software development, or the quality system of the laboratory that uses the software.

The degree to which security issues are included also varies: Some Guides include them, other refer to existing standards.

Additionally, it became evident that none of the documents reviewed in the survey are applicable to the problem in hand. The following may be set out as the most important reasons:

- improper definition and refinement of testable requirements,
- improper validation guidance, both because of selected methods and understandability of application for target audience, and
- improper addressing of implemented IT functionality.

5 Development of the WELMEC 7.2:2005 – Software Guide [7]

The initial research work for the core subject of the present thesis, i.e. the validation procedure for metrological software, was conducted during the development phase of the WELMEC Guide 7.2.

Both direct and indirect software requirements of the MID considerably vary in their complexity and technical realisation. As far as legal metrology is concerned, constraints to the features of the instrument differ in various areas of application. It is the idea of the newly developed Guide to reflect these given facts through a modular structure. The Guide consists of two basic requirement sets, which have to be applied alternatively: one set is intended for simple, built-for-purpose measuring instruments (set P), and the other has to be applied to more complex systems containing a universal computer (set U). Depending on special features, which are typical for contemporary measuring instruments, these basic requirement sets have to be extended by additional requirements: the Guide contains specific requirements on long-term storages, secure transmission via communication networks, separation of legally relevant and legally non-relevant software parts, and software download. Besides these general enhancements, the Guide contains additional software requirements that are specific for each type of instrument covered by the MID. These requirements have been derived from the instrument-specific Annexes of the MID.

In practice, a designer or an examiner of a software-controlled measuring instrument has to decide which of the requirement modules are to be applied to a certain instrument. He selects one of the basic requirement sets, up to four additional sets for special technical features, and possibly one further set specific for the respective kind of instrument.

Besides the technical aspects, the Guide 7.2 takes into consideration the differences in the fields of application of measuring instruments, by defining the „risk classes“. Here, the risk of fraud, the risk of insufficient examination, and possible deficiency of conformity, are combined into risk classes (in contrast to the risk classes in safety). The requirement sets described above are sub-divided into sets with different severity corresponding to the risk classes. For each requirement of the Guide, validation steps and examples of technical solutions are proposed that suit the particular risk class. The requirements do not necessarily differ between one risk class and another. In many cases they are identical. The differences between risk classes emerge again in different technically acceptable solutions, e.g. examples of how a requirement can be fulfilled.

In previous sections, information about the technical, legislative and application environment for metrological software application has been presented. This section explains the preparation and contents of concrete validation guidance for software components placed in such environment.

5.1 Essential requirements – the starting point

The software validation Guide needed to give straightforward instructions for validation of the following software-related requirements in MID, Annex I:

7. Suitability

7.6

A measuring instrument shall be designed so as to allow the control of the measuring tasks after the instrument has been placed on the market and put into use. If necessary, special equipment or software for this control shall be part of the instrument. The test procedure shall be described in the operation manual.

When a measuring instrument has associated software which provides other functions besides the measuring function, the software that is critical for the metrological characteristics shall be identifiable and shall not be inadmissibly influenced by the associated software.

8. Protection against corruption

8.1

The metrological characteristics of a measuring instrument shall not be influenced in any inadmissible way by the connection to it of another device, by any feature of the connected device itself or by any remote device that communicates with the measuring instrument.

8.3

Software that is critical for metrological characteristics shall be identified as such and shall be secured.

Software identification shall be easily provided by the measuring instrument.

Evidence of an intervention shall be available for a reasonable period of time.

8.4

Measurement data, software that is critical for measurement characteristics and metrologically important parameters stored or transmitted shall be adequately protected against accidental or intentional corruption.

8.5

For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use.

Being rather general, these requirements needed refinement in terms of software testability. On the other hand, since conformity assessment of legal measuring instruments is supposed to be performed by metrologists rather than software testing experts, such refined requirements needed appropriate validation guidance. WELMEC Guide 7.2 is the result of the endeavours to meet these – at first glance opposite – prerequisites.

For validation purposes, these requirements have to be interpreted in order to be applicable to software and legal metrology validation and verification.

Certain international standards and guidance documents, covering particular, narrow segments of the identified problem from several points of view, already existed. The decision was to draw up a specific, targeted metrological software validation guidance document with interdisciplinary synthesis from existing sources as the starting point. The fundamentals for preparation of the guide were based on the following requirements and technical documents, as illustrated in Figure 5-1:

- Domain-specific regulations and standards, e.g.:
    - Directive on Measuring Instruments 2004/22/EC (MID) [4];
    - OIML R 49-1 – EN: 2006 Water meters intended for the metering of cold potable water and hot water. Part 1: Metrological and technical requirements (and many other OIML recommendations) [35];
    - EN 50470-1:2006 CLC/TC 13 Electricity metering equipment (a.c.) –Part 1: General requirements, tests and test conditions – Metering equipment (class indexes A, B and C) 6060 2004/108/EC, 2004/22/EC - [6];
    - EN 50470-3:2006 CLC/TC 13 Electricity metering equipment (a.c.) – Part 3: Particular requirements – Static meters for active energy (class indexes A, B and C) 6060 2004/22/EC [8]; and
    - ISO/IEC 17025: 2005 General requirements for the competence of testing and calibration laboratories) [21].

  These documents address wide metrological aspects of measuring instruments, while addressing the software barely or not at all.
- Software product quality standards (e.g. ISO/IEC 9126-1÷4 Software engineering –Product quality [23]).
- Software product/process validation standards, e.g.:
    - ISO/IEC 12119 Information technology – Software packages – Quality requirements and testing [17];
    - ISO/IEC 15504-1÷5 Information technology – Process assessment [20] ;and

  - ISO/IEC 15408-1÷3 Information technology – Security techniques – Evaluation criteria for IT security [19].

- software validation methods and strategies, e.g.:
  - IEC 61508-1÷7 Functional safety of electrical/electronic/programmable electronic safety-related systems [9];
  - Glenford Myers "The Art of Software Testing" [28] and BS 7925-2: The Software Component Testing [2].

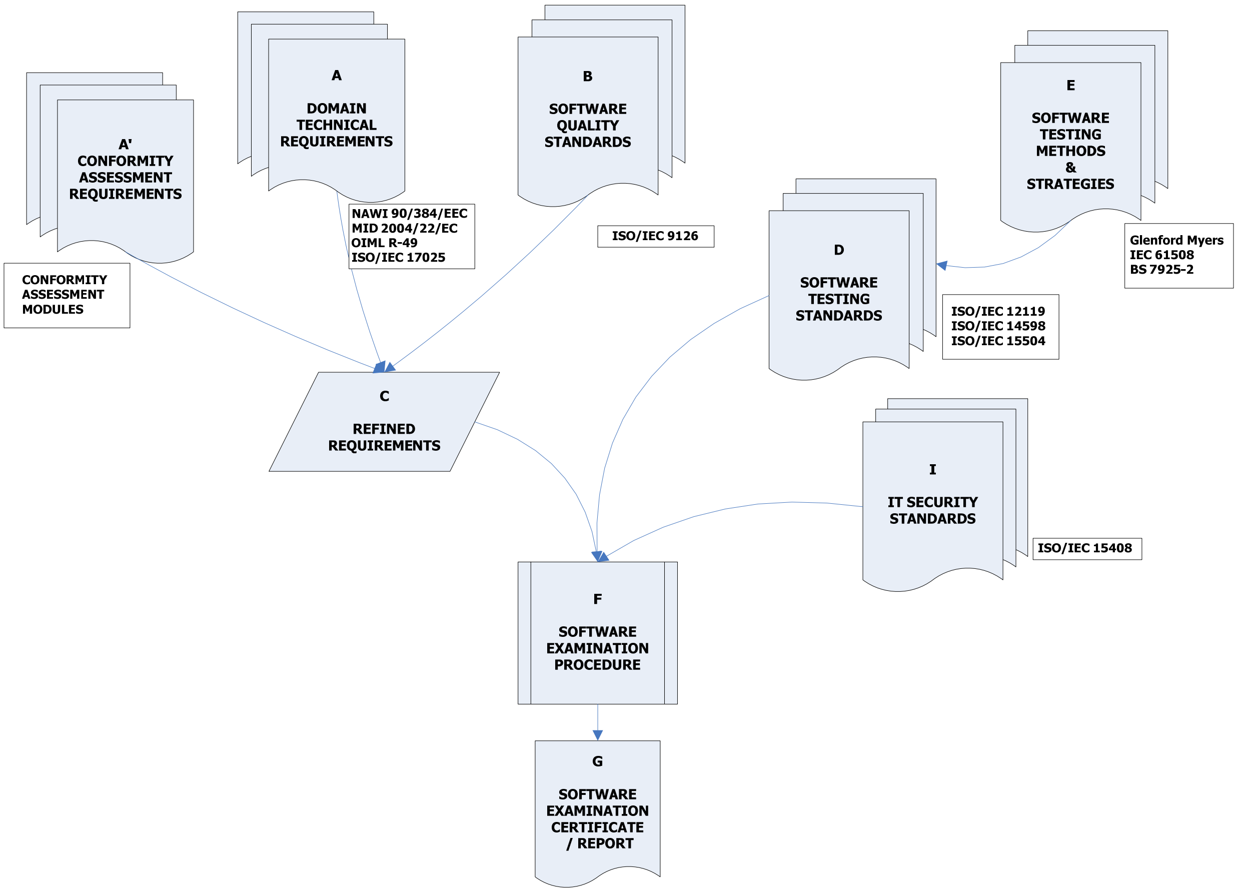

A' CONFORMITY ASSESSMENT REQUIREMENTS
CONFORMITY ASSESSMENT MODULES
A DOMAIN TECHNICAL REQUIREMENTS
NAWI 90/384/EEC
MID 2004/22/EC
OIML R-49
ISO/IEC 17025
B SOFTWARE QUALITY STANDARDS
ISO/IEC 9126
E SOFTWARE TESTING METHODS & STRATEGIES
Glenford Myers
IEC 61508
BS 7925-2
D SOFTWARE TESTING STANDARDS
ISO/IEC 12119
ISO/IEC 14598
ISO/IEC 15504
C REFINED REQUIREMENTS
I IT SECURITY STANDARDS
ISO/IEC 15408
F SOFTWARE EXAMINATION PROCEDURE
G SOFTWARE EXAMINATION CERTIFICATE / REPORT

Figure 5-1: Interdisciplinary elements included in the synthesis of the metrological software validation guidance

As an outcome, the document WELMEC 7.2:2005 – Software Guide (Measuring Instruments Directive 2004/22/EC) was elaborated. Considering its structure and intended use, this guidance document may be treated as the skeleton of a forthcoming international standard document for examination of software in measuring instruments and systems.

5.2 Overview of the developed guidance document

The newly developed Guide provides support to all those concerned with the application of the MID. It addresses both the manufacturers of measuring instruments and third-party examiners. The guide is of a purely advisory nature, and does not itself impose any restrictions or additional technical requirements beyond those contained in the MID. Alternative approaches may be acceptable, but the guidance provided in this document represents the best practice to be followed. Although the Guide is oriented to instruments included in the regulations of the MID, the results are of a general nature and may be applied wider.

The main content of the Guide consists of specific software requirements based on the essential general requirements as given in the MID Annex I. Furthermore, recommendations are given to carry out validations of software.

The overall structure of the requirements follows the classification into configurations of measuring instruments. There are so-called "basic" configurations and "extended" IT configurations. The basic configurations are divided into two classes:

- built-for-purpose measuring instruments with embedded IT components and dedicated application software (called P type instruments), and
- measuring instruments using a universal computer and software running on it (called U type instruments).

Each type of a measuring instrument can be assorted to exactly one basic configuration. The basic rule is that each instrument that cannot be unambiguously classified as a P type instrument is a U type instrument.

The extended IT configurations are associated with typical IT functions, e.g.:

- long-term storage of measurement data (called extension L),
- transmission of measurement data (called extension T),
- software download (called extension D),
- software separation (called extension S).

Each set of these requirements is only applicable if the corresponding function exists.

Furthermore, risk classes have been introduced. All requirements are differentiated according to risk classes. This means that before a measuring instrument can be assessed, the applicable risk class must be chosen, so that the appropriate requirement blocks can be selected. Risk for software in legal metrology is influenced by three factors:

- risk of accidental changing or tampering software and measurement data ,
- risk of individual instruments not conforming to the approved type, and

- risk of type approval examinations not being exhaustive enough to reasonably reduce the probability of deviations from requirements.

The definitions in WELMEC Guide 7.2 are organised as a structured set of requirement blocks (refer to Figure 5-2). The overall structure follows the classification of measuring instruments into basic configurations and the use of IT configurations, as described above. These sets of requirement blocks are complemented by instrument-specific requirements. The instrument-specific requirements cannot be considered isolated from the instrument-independent parts. They are restricted to specific aspects of measuring instruments.

The use of the Guide is supported by a recommended procedure. This procedure proposes the following steps:

- Step 1: Selection of basic configuration (P or U);
- Step 2: Selection of applicable IT configurations (extensions L, T, S and D);
- Step 3: Selection of instrument-specific requirements (extension I);
- Step 4: Selection of applicable risk class.

In addition to this procedure, checklists have been introduced to further support work with the Guide. The checklists are the means to ensure that all the requirements within a chapter have been covered by the manufacturer or third party examiner.

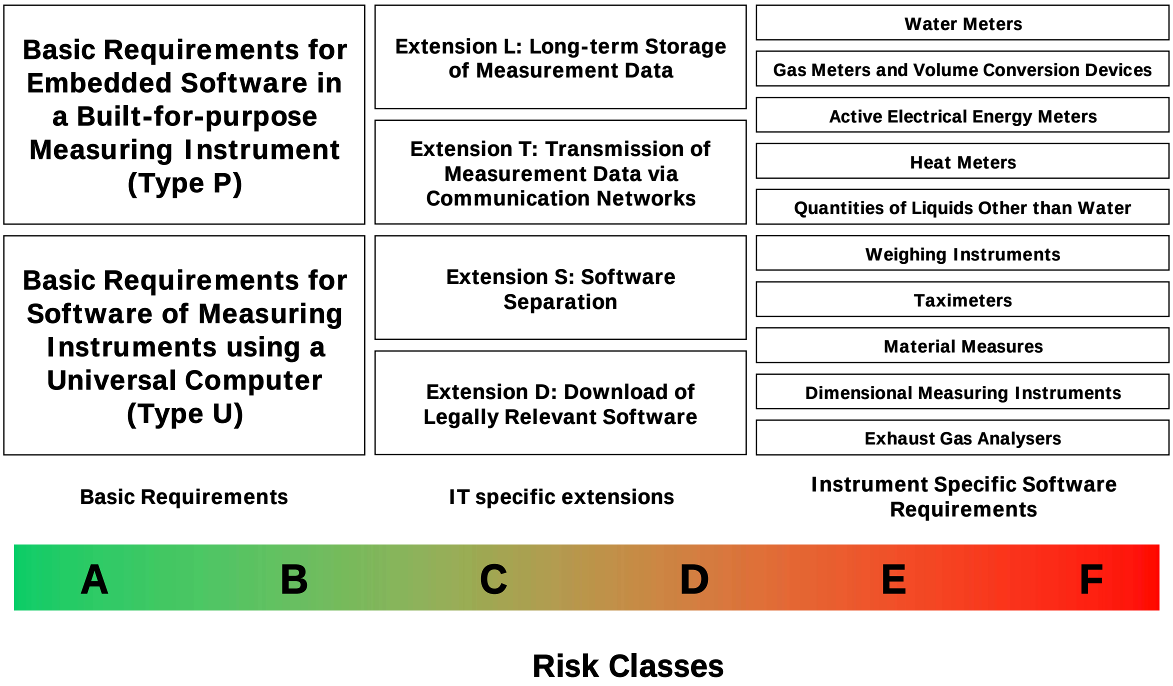

Basic Requirements for Embedded Software in a Built-for-purpose Measuring Instrument (Type P)
Basic Requirements for Software of Measuring Instruments using a Universal Computer (Type U)
Extension L: Long-term Storage of Measurement Data
Extension T: Transmission of Measurement Data via Communication Networks
Extension S: Software Separation
Extension D: Download of Legally Relevant Software
Water Meters
Gas Meters and Volume Conversion Devices
Active Electrical Energy Meters
Heat Meters
Quantities of Liquids Other than Water
Weighing Instruments
Taximeters
Material Measures
Dimensional Measuring Instruments
Exhaust Gas Analysers
Basic Requirements
IT specific extensions
Instrument Specific Software Requirements
A
B
C
D
E
F
Risk Classes

Figure 5-2: Modular structure of the WELMEC Guide 7.2

To support the validation of requirements by independent test authorities, validation recommendations have been developed and associated with the requirements. These recommendations are differentiated according to the risk classes. Furthermore, a set of test methods for the validation has been compiled. This compilation is not part of the requirement guide, but an additional means of support.

5.2.1 Characteristics of the instruments considered by particular modules

5.2.1.1 Basic configuration P

A P type instrument is roughly characterised by the following features:

- The entire application software has been designed for the purpose of measuring.
- The software is designed and treated as a whole, unless software separation according to Extension S has been observed.
- The user interface is exclusively intended for the purpose of measuring.
- There is no operating system having a user shell that is independent from measurement.
- The software and its environment are invariable, and there are no means for programming or changing the software by the user. Software download is only allowed if Extension D is observed.

5.2.1.2 Basic configuration U

The instrument fits into this category if at least one of the conditions of a P type instrument is not fulfilled. Typically, the following features characterise a U type instrument:

- A modular general-purpose computer system is used. The computer system may be stand-alone, part of a closed network, or part of an open network.
- The sensor is normally external to the computer unit and linked to it by a communications link.
- The user interface may be switched from an operating mode, which is not under control of the measuring process, to one which is, and vice-versa.
- Any operating system may be used. In addition to the measuring instrument application, other software applications may also reside on the system at the same time.
- The software and its environment are invariable and there are no means for programming or changing the software by the user. Software download is only allowed if Extension D is observed.

5.2.1.3 Extended IT configurations are associated with typical IT functions:

Each set of requirements related to extended IT configurations is only applicable if the corresponding functionality is implemented in the instrument.

Extension L describes the requirements for the storage of measurement data from the moment when a measurement is physically completed to the point in time when all measurement processes to be done are finished. It may also be applied to long-term storage of the data thereafter.

Extension T must be used when measurement data are transmitted via communication networks to a distant device, where they are further processed and/or used for measurement purposes.

Extension S deals with software separation. Software separation is a kind of modular design methodology adapted in particular to legal metrology. Software-controlled measuring instruments or systems in general have complex functionality and contain modules that are legally relevant, and modules that are not. It is advantageous for manufacturers to separate these software modules of the measuring system in order to easily modify non-legally relevant software.

Extension D shall be used for downloading software to measuring instruments. The requirements are to be considered when software or parts of the software are to be replaced by another version. The aim of these requirements is to ensure that the integrity and authenticity of software to be downloaded are not affected, and that the functionality and security characteristics of the instrument do not change, except those that are meant to be changed by the download.

5.2.2 Risk classes

Software validation requirements are differentiated according to the estimated risk of malfunction of the instrument, which is determined by the area of intended use of the instruments (e.g. potential health hazard in the case of malfunction of an instrument used in medical diagnostics, or measurement of environmental conditions, or potential financial fraud in measuring instruments used in commercial transactions). This means that, before a measuring instrument can be assessed, the applicable risk class must be determined, so that the appropriate requirement blocks can be selected.

Risk levels for the relevant aspects of software protection, software conformity and software examination are defined below.

Software protection levels

- Low: No particular protection measures against intentional changes of software and data are required.
- Middle: The software and the data are protected against intentional changes carried out by third-parties using easily available and simply usable software tools (e.g. a text editor, common database access means).
- High: The software and the data are protected against intentional changes carried out by third-parties using sophisticated tools (debuggers and hard disc editors, software development tools, etc.).

Software conformity levels

The required conformity of the software of an instrument in serial production with the software of the instrument submitted to type evaluation may be:

- Low: The functionality of the software implemented for each individual instrument is in conformity with the documentation of the software for the pattern of the approved instrument.
- Middle: In addition to the low conformity level, depending on the technical features, the critical parts of the software (source code) may be defined as fixed. The fixed part shall be identical in every individual instrument.
- High: The software (source code) implemented in the individual instruments is completely identical to the software implemented in the pattern of the approved instrument.

Software examination levels

- Low: No extra software testing is performed. A functional test of the instrument containing the software is carried out.
- Middle: In addition to the low level, the documentation of the software is checked, and practical tests of the particular functions that are software-supported (spot checks) are carried out on the instrument.
- High: In addition to the medium level, an in-depth test of the software is carried out, usually based on the source code.

Out of the 27 theoretically possible level permutations, only 6 are of interest, because of the relations between the different categories that exclude arbitrary combination. In addition, they cover all of the instrument classes considered so far. Moreover, the risk classes provide a sufficient window of opportunity for the case of changing risk evaluations. The classes are presented in Table 5-1. They may be interpreted as follows:

- Risk class A: It is the lowest risk class of all. No particular measures are required against intentional changes of software and data. Validation of software is part of functional testing of the instrument. Conformity is required at the level of documentation.
- Risk class B: In comparison to Risk class A, the protection of software is required at the middle level. Correspondingly, the examination level is raised to the middle level. Conformity remains unchanged in comparison to Risk class A.
- Risk class C: In comparison to Risk class B, the conformity level is raised to middle. This means that parts of the software may be declared as fixed. The rest of the software is required to conform to the functional level. The levels of protection and examination remain unchanged in comparison to Risk class B.
- Risk class D: The significant difference in comparison to Risk class C is the raising of the protection level to high. Since the examination level remains unaffected at "middle", sufficiently informative documentation must be provided to show that the protection measures taken are appropriate. The conformity level remains unchanged in comparison to Risk class C.
- Risk class E: In comparison to Risk class D, the examination level is raised to high. The levels of protection and conformity remain unchanged.
- Risk class F: The levels with respect to all aspects (protection, examination and conformity) are set to high.

Risk Class	Software Protection	Software Examination	Degree of Software Conformity
A	*Low*	*Low*	*Low*
B	*Middle*	*Middle*	*Low*
C	*Middle*	*Middle*	*Middle*
D	*High*	*Middle*	*Middle*
E	*High*	*High*	*Middle*
F	*High*	*High*	*High*

Table 5-1: Risk classes, SW protection, examination and conformity levels

5.2.3 Instrument-specific requirements

The definitions in WELMEC Guide 7.2 are organised as a structured set of requirement blocks. The overall structure follows the classification of measuring instruments into basic configurations and the use of IT configurations, as described in the previous sections. These sets of requirement blocks are complemented by instrument-specific requirements. The instrument-specific requirements cannot be considered isolated from the instrument-independent parts. They are restricted to specific aspects of measuring instruments.

The collection of instrument-specific requirements is called “extension I” in WELMEC Guide 7.2, Extension I is subdivided according to the classes of measuring instruments, and has an open structure, e.g. it provides a skeleton that is filled-in only if appropriate. There are different instrument-specific software aspects that might need specific consideration for a certain type of measuring instrument. Typically, only a few or even no specific requirements are necessary for a certain class of instruments. The intention was to cover the following additional aspects:

- software requirements induced by specific regulations, standards and normative documents,
- software requirements derived from typically used technical configurations or set-ups,
- software requirements derived from the measuring principle,
- software fault detection and reaction that is required by specific types of instruments,
- software requirements reflecting a specific hardware configuration,
- protection measure for specific parameters, and
- specific documentation required.

In the current issue of the WELMEC Guide 7.2, the following groups of instruments are covered:

- water meters,

- gas meters and volume conversion devices ,
- active electrical energy meters,
- heat meters,
- measuring systems for the continuous and dynamic measurement of quantities of liquids other than water,
- weighing instruments,
- taximeters, and
- exhaust gas analysers.

5.3 An example of application of the WELMEC Guide 7.2

In the following subsection an application of the WELMEC Guide 7.2 to a specific unit under test is described, step by step. The selected type of instrument is a static meter of active electrical energy.

The validation procedure includes the following steps:

- Determination of applicable requirements:
  - decision on instrument type,
  - determination of IT configuration (applicable extensions),
  - determination of risk class,
- Performance of the validation
- Preparing the report

Decision on instrument type is supported by the checklist in Table 5-2 below:

	Questions for decision on instrument type	**Answers that meet the conditions for a P – type instrument**
1.	Is the entire application software designed for the measuring purpose?	(Y)
2.	If there is general-purpose software, is it accessible by or visible to the user?	(N)
3.	Is the user prevented from accessing the operating system, when it is possible to switch to an operating mode not subject to legal control?	(Y)
4.	Are the implemented programs and the software environment invariable (apart from updates)?	(Y)
5.	Are there any means for programming?	(N)

Table 5-2: Decision on instrument type

In the considered case study, the unit under test met the conditions to be classified as a type P instrument.

Decision about extended IT configurations is supported by the checklist in Table 5-3.

	Questions for decision on extended IT configurations	**YES**	**NO**	**N/A.**
L	Does the device have the ability to store the measurement data either on an integrated storage or on a remote or removable storage?	x[1]		
T	Does the device have interfaces for transmission of data to devices subject to legal control, OR is the device receiving data from another device subject to legal control?		x	
S	Are there software parts with functions not subject to legal control, AND are these software parts desired to be changed after type approval?		x	
D	Is downloading of software possible or desired?		x	

Table 5-3: Decision about extended IT configurations

The software contains functions which are subject to legal control as well as other functions. However, the software is handled and treated as a whole. Therefore Extension S does not apply. Extensions T (transmission of data) and D (software download) do not apply.

Instrument-specific requirements – Extension I3 for active electric energy meters applies.
Risk class C has been assigned to the active electric energy meters of type P.

A complete list of requirements and validation guidance applicable to built-for-purpose meters of active electric energy is given in Table 10-1 (Section 10.1).

[1] The cumulative energy register isn't considered as long-term storage in the sense of Extension L. Therefore Extension L applies to the stored value profile only.

6 Verification of interconnection between the WELMEC Guide 7.2 and the related fields

A very important part of approval of the newly developed software validation guidance document was the verification of its proper interconnection with all the related fields. The necessity arose from the concern that the WELMEC Guide 7.2 might be in contradiction with some legislative, normative or guidance document from the related fields. This could have been the case, given the fact that such systematic approach did not exist before, and other related areas could have been slightly divergent. Additionally, it was necessary to confirm that the new guide was in line with the state of the art. Linkage with the following was checked:

- Domain (metrological) standards (e.g. the Directive on Measuring Instruments);
- Normative documents related to metrological software and metrological software guidance documents (e.g. WELMEC and OIML guidance documents);
- International software quality standards (e.g. ISO/IEC 9126);
- International software testing standards (e.g. ISO/IEC 12119), and
- Conformity assessment requirements.

6.1 Linkage between generic software quality requirements and requirements originating from legal metrology [13]

The common points between the WELMEC Guide 7.2 and the International Standard ISO/IEC 9126:2001-1 are presented in Table 10-2, in Section 10. Figures in Table 10-2 represent the number of requirements in the particular sections of the WELMEC Guide 7.2 concerning the examined measuring instruments (i.e., for electric energy meters: P, U, L, T, S, D and I-3) that correspond to the particular software quality requirement of the International Standard ISO/IEC 9126-1:2001.

The weighting factors in Table 10-2 are summarised for each quality characteristic and subcharacteristic. According to the objectives of the MID and the WELMEC Guide 7.2, the focus of the MID requirements is targeted on the quality characteristic Functionality, in particular the sub-characteristic Security, and on a limited scale on Reliability. This is one reason for the high weighting factors of Functionality (61) and Security (33). However, there is another reason for the high weighting factor of Functionality. Many of the requirements are system- rather than software-related requirements. Such requirements were assigned to the quality characteristic Functionality, in particular to the sub-characteristic Suitability. In accordance with the MID objectives, it is obvious that Efficiency is not an important characteristic.

All comprising, the definitions of the WELMEC Guide 7.2 requirements have reached a sufficient coverage of the ISO/IEC 9126 quality characteristics concerning the following attributes:

- Functionality, in particular security,

- Reliability, and
- Maintainability and Portability (in relation to the importance within MID).

An insufficient coverage was ascertained only in the case of the quality characteristic Usability. In particular, there is a lack of explicit requirements regarding software documentation.

6.2 Linkage between the generic software testing standard "ISO/IEC 12119:1994 Information technology – Software packages – Quality requirements and testing" and the validation guidance prepared for legal metrology conformity assessment

The aim of this analysis was to determine the level of equivalence between the requirements of the WELMEC Guide 7.2 and the generic software testing standard "ISO/IEC 12119:1994 Information technology – Software packages – Quality requirements and testing". An overview is presented in Table 10-3, in Section 10. The outstanding issues can be summarized as follows:

a) Issues considered more important by WELMEC Guide 7.2 than by ISO/IEC 12119:

- WELMEC Guide 7.2 addresses security much more in-depth than ISO/IEC 12119, especially functional testing of security.

b) Issues not covered by WELMEC Guide 7.2:

- WELMEC Guide 7.2 does not address the quality of user documentation. It is only mentioned as a necessary part of the documentation,
- Installability, efficiency, maintainability and portability are not covered at all.

c) Other comments:

- Potential software failures are treated from the point of view of functionality (security) rather than reliability;
- Correctness, consistency, reliability and usability are mentioned only in connection with the extended IT configurations (and not for basic configurations P and U);
- Testing and documentation of interfaces are suitably addressed;
- Boundary values analysis is the only explicitly mentioned testing strategy.

6.3 Extracting the relevant requirements and approaches from existing international normative documents related to metrological software, and worldwide guidance documents related to metrological software

This analysis has been performed with the aim to overview the guidance documents applicable to the software of measuring instruments, and to make a rough comparison of their approaches and contents.

Although in principle focused on the same domain, it appears that the analysed documents cover rather different issues related to metrological software. The main difference in the guidance documents is their focus. The majority of guidance documents address the software in measuring instruments or systems, while some guidance documents address the laboratory administrative software (including office documentation software), quality system management software and laboratory information systems as well.

The software in measuring instruments and systems is mostly addressed trough the product approach, while some guidance documents treat the software lifecycle processes. In addition, it became obvious that there existed different groups of intended users:

- manufacturers of measuring instruments,
- conformity assessment experts at various stages (e.g. design and development, type approval and field use), and
- metrological experts in laboratories, who develop measuring software applications by themselves.

Validation guidance in the analysed documents varies both in comprehension and level of expertise, furthermore, two of the documents do not contain validation guidance at all.

The intention to deal with risk assessment and its consideration is present in almost all the guidance documents. The assessment of risks and their consequences (e.g. the determination of “risk class” or similar categorisation) varies with regard to the different risks involved in application of the software (stricter in medical devices, less strict in legal metrology or testing and calibration, however, with exceptions, e.g., in forensic tests). The degree to which security issues are included also varies: some guides include them, other refer to existing standards.

As a general conclusion, we may consider that the different approaches in the requirement and validation guidance documents that have been analysed, come from the priorities of the particular sub-domain. A general observation was that every document was developed spontaneously, whenever a certain community faced a particular problem.

More details on this analysis are available in Section 4 and [37]. An overview of the requirements and validation suggestions is presented in Tables 10-4 and 10-5, in Section 10.

6.4 Binding the areas of conformity assessment in legal metrology with the international practice of software testing

As already explained in Section 1.4 herein, according to the "New Approach", conformity assessment of a measuring instrument may be performed in several ways (or modules). Conformity assessment activities are distributed between the manufacturer, the notified body and the notified body surveillance institution.

The following conformity assessment modules may be used for the software-controlled measuring instruments or systems regulated by the Directive on Measuring Instruments:

- B – Type examination;
- D – Declaration of conformity to type based on quality assurance of the production process;
- F – Declaration of conformity to type based on product verification;
- G – Declaration of conformity based on unit verification; and
- H1 – Declaration of conformity based on full quality assurance plus design examination.

An overview of the link between the particular conformity assessment modules and software testing activities (under the responsibility of a manufacturer or notified body) during MID-conformity assessment of software-controlled instruments is presented in Table 10-6, in Section 10.

Applicable combinations of the conformity assessment modules are B+F, B+D, H1 or G.

At this stage, the validation guidance presented in this thesis covers conformity assessment procedures related to module B. Extension to modules H1 and G is a matter of future work.

More details about the activities and issues in different modules of the software-related part of conformity assessment of measuring instruments are available in [36].

6.5 Establishing links between domain-specific standards for software components used in metrological applications, generic software quality standards, software testing standards, software testing methods and software testing strategies

Practical implementation of a technical regulation requires a refinement of requirements from “high-order” functional requirements (as stated in a directive, e.g. the MID), through domain-specific amendment of requirements for software components (in this particular case, the WELMEC Guide 7.2), to the validation guidance based on state-of-the-art software testing approaches adopted for application by the intended user (with explanations on practical examples) [26].

Although the software testing knowledge base and praxis offer a very wide range of methods, strategies, tools and techniques, it became evident, as an outcome of this analysis, that a major part of validation can be done by rather simple means (see Figure 6-1 below). For example, the analysis of user documentation covers approximately 50% of validation, especially for lower-risk classes. Advanced approaches will be used very rarely, e.g. for validation of software of measuring instruments in higher-risk classes.

The connection between domain-specific standards and the newly developed guidance is illustrated in Table 10-7 (Section 10) on an example of relation between the requirements of MID and the refined requirements of WELMEC Guide 7.2 [26].

An overview of the proposed validation methods per individual requirement is presented in Table 10-8 (Section 10). Additionally, Table 10-9 gives a simplified overview of suggested methods for validation of individual WELMEC Guide 7.2 requirements. An interesting finding is that a major part of validation can be performed by non-sophisticated methods. Source code analysis, as the most stringent method, appears only in issues that regard security in the higher-risk classes.

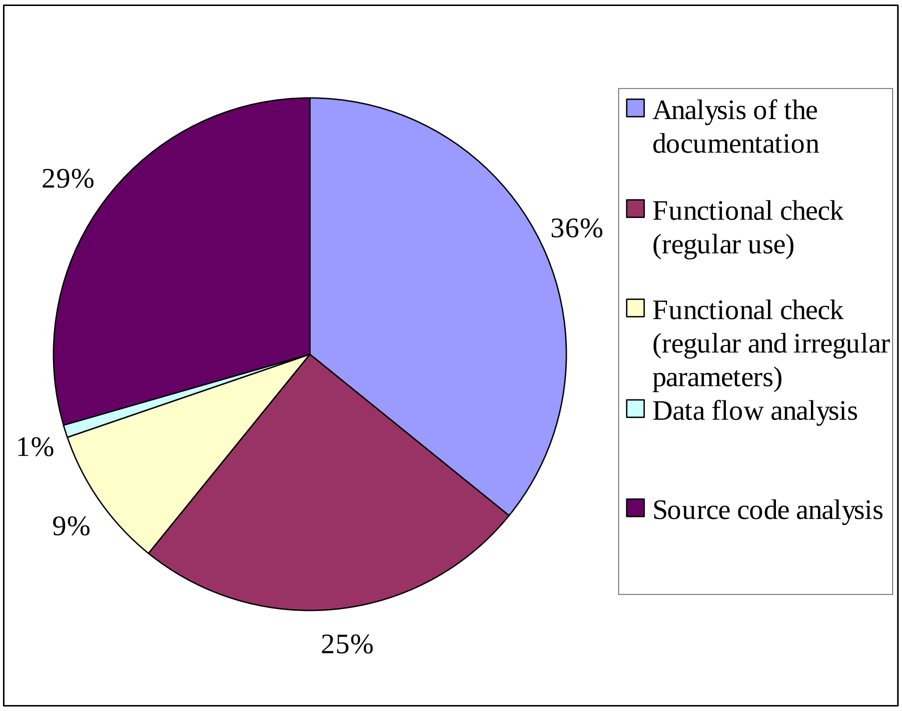


Figure 6-1: Suggested validation methods by WELMEC Guide7.2

7 Comparative validation of a measuring instrument's software – performance, outcome and follow-up activities

7.1 Motivation for the experiment

Both professional and scientific approaches require for every newly developed procedure to be thoroughly validated before first use, in order to confirm its value and usability. In this case study a new procedure to be validated was the guidance document WELMEC 7.2:2005 – Software Guide [41] (Measuring Instruments Directive 2004/22/EC [4]). The main objective of the WELMEC Guide 7.2 was to support harmonised conformity assessment of the software in measuring instruments covered by the MID between all institutions performing conformity assessment for the European market (as explained in Section 1.4). Therefore it was necessary to ensure that the findings of the validation of the same software component were the same, regardless of the institution which performed the validation. Consequently, understanding of the requirements and application of the validation procedures among different performers of software validation had to be the same.

As the most appropriate method for validation of the Guide emerged an experiment in which several laboratories perform validation of the same software component according to the requirements and suggested validation methods from WELMEC Guide 7.2. The experiment was named "comparative examination".

Although elaborated quite comprehensively (taking into account technological realisation, implemented functionality and the specifics of intended use of a measuring instrument – “risk classes”), the software validation guidance in the document WELMEC Guide 7.2 gives the performer of testing a significant freedom based on his engineering knowledge, especially in terms of selection of test methods and strategies to be applied. Consequently, there existed a latent danger of different interpretation between various performers of software validation, which could result in obtaining different test results and conclusions on conformity for the same unit under test.

The concern about possible different understanding and application of the elaborated Guide arose among the participants of the MID-SOFTWARE network (representatives of both manufacturers and national metrology institutes) already during the preparation of the guidance document. On the other hand, the awareness of the necessity of testing the equivalence of approaches of the performers of conformity assessment of metrological software through comparative validation of the measuring instrument's software (called also "round robin"), had already been present for some time within WELMEC WG7 [53], but has not yet been realised. The implementation of the MID was an ideal opportunity for carrying this out, especially having in mind that guidance document for software validation was already prepared at that time. The main uncertainty which had to be resolved was the concern whether the WELMEC Guide 7.2

was clear and straightforward enough to ensure that testers from different laboratories would come to same results when examining the same software component. Members of WELMEC WG7 decided to resolve this constraint before October 30th, 2006 – the date of entering into force of the MID.

7.2 Performance of validation of WELMEC Guide 7.2

The goal of the validation of the Guide was to determine the degree of equivalence of approaches to validation of software components in legal metrology between different national legal metrology authorities.

Laboratories from six European national metrology institutes took part in the intercomparison experiment [88]. The selected unit under test was the software of an electrical energy meter, which is regulated under the Directive on measuring instruments. The units under test, together with the accompanying equipment and documentation, were provided by Landis+Gyr[1]. The experiment was co-ordinated by the Metrology Institute of the Republic of Slovenia (MIRS).

The test plan was prepared according to the validated document WELMEC Guide 7.2:2005. The basic principles, as well as the organisational approaches were assumed from the ISO/IEC GUIDE 43-1:1997 [25] and BIPM Guidelines for CIPM key comparisons [1].

The equipment under test (EUT) was an active meter of electrical energy without communication interfaces and without the functionality of remote software download, type ZxD100AR, manufactured by Landis+Gyr Ltd. Software validation was performed as part of the type examination procedure (conformity assessment according to the module "B").

The assumed risk class was “C”. The EUT documentation was in English. One EUT package was sent to each participant in the exercise. This made it possible for all participants to perform the work simultaneously. The EUT packages had been prepared and shipped by the manufacturer.

Each EUT package contained:

- one Meter Landis+Gyr Ltd. ZxD100AR,
- a communication cable (for IR communication according to IEC 62056-21/DLMS [15]),
- Landis+Gyr MAP110, communication/parametrisation software for PC together with the necessary user authentication data,
- H 71 0015 0029 en - ZMD100AR - MID Software Declaration.pdf,
- H 71 0200 0268 en - ZMD100AR - User Manual.pdf,
- H 71 0200 0270 en - ZMD100AR - Functional Description.pdf,
- H 71 0200 0405 en - ZMD100AR - Software Description.pdf,
- H 71 0200 0406 en - ZMD100 AR - Parameter List.xls,

[1] http://www.landisgyr.com/index.cfm

- H 71 0200 0332 en - MAP110 - User Manual.pdf,
- MAP_07en_Conv_of_ B14Parameterisation_Trees.pdf,
- MAP_15 en - Optical Port Settings of ZxQ Meters.pdf,

7.2.1 Overview of the steps:

The agreed project steps and timeframe were the following:

Phase	Deadline
Preparation of draft instructions for performance of the work (MIRS)	6 June 2006
Collection of comments to the draft instructions for performance of the work (by the members of WELMEC WG7 [53])	20 June 2006
Distribution of improved instructions (MIRS)	23 June 2006
Distribution of the EUT[2] packages to all participants in the experiment (Landis+Gyr)	23 June 2006
Performance of the validation (all participants)	
Collection of test results on agreed format (MIRS)	6 October 2006
Analysis of the results (GAR[3])	23 and 24 October 2006
Discussion of the results[4]	25 October 2006
Preparation of report on the experiment (GAR)	30 October 2006

7.2.2 Course of the experiment

7.2.2.1 Objectives of the exercise

The main objective of the exercise was to validate the application of the WELMEC Guide 7.2 between the institutions who perform conformity assessment of the MID MI-003 instruments according to module B. It was not meant to be a competition between the laboratories, but a validation of their understanding and the applicability of the Guide.

Nevertheless, from the results and intermediate data it was possible to extract valuable information for the improvement of the work of each individual participant.

7.2.2.2 Rules of performance

The following rules were agreed between the participants of the experiment:

[2] EUT: Equipment Under test

[3] GAR: Group for Analysis of the Results: Christoph Rahm (Landis+Gyr), Paul Kok (NMi), Dieter Richter (PTB), Tanasko Tasić (MIRS) and Roman Flegar (MIRS).

[4] The initial idea was to organise a separate, second meeting of the experts who performed the validation.

1. Every participant in the exercise had to identify the requirements from the WELMEC Guide 7.2 independently.
2. Validation had to be performed according to the validation guidance in the WELMEC Guide 7.2. The selection of test methods and strategies (analysis of the documentation, functional checks, dynamic black-box testing, or other – as explained in Section 3) was the choice of each individual participant.
3. The suggested procedure for the performance of the work in the laboratory was:
   - Identification of requirements;
   - Selection of methods for checking the requirements;
   - Preparation of detailed test plan including the selected test environment, test methods and test cases;
   - Performance of the validation;
   - Preparation of test report;
   - Sending the test report to MIRS.
4. In the case of detecting a bug during the testing:
   - The failure had to be recorded;
   - EUT's software or EUT itself was not supposed to be replaced by a new EUT;
   - The testing had to be continued.

7.2.2.3 Outputs of the testing process by each participant

Each participant had to provide the following documents as the outcome of the testing process:

- test plan,
- test report (the skeleton of the test report is defined in Chapter 12 of the WELMEC Guide 7.2.),
- text to be included in the Type Approval Certificate (as defined in Chapter 12.4 of the WELMEC Guide 7.2),
- short evaluation report containing comments on possible ambiguities, the applicability of particular parts of the Guide 7.2 and recommendations for enhancements,
- an overview of the testing process:
    - requirement,
    - test method,
    - result,
    - comment.

The reference set of applicable requirements for the selected unit under test expected to be identified by every participant in the experiment was as follows:

P1 – Documentation

P2 – Software identification

P3 – Influence via user interface

P4 – Influence via communication interface

P5 – Protection against accidental or unintentional changes

P6 – Protection against intentional changes

P7 – Parameter protection

L1 – Completeness of measurement data stored

L2 – Protection against accidental or unintentional changes

L3 – Integrity of data

L4 – Authenticity of measurement data stored

L5 – Confidentiality of keys

L6 – Retrieval of stored data

L7 – Automatic storing

L8 – Storage capacity and continuity

I3-1 – Fault Recovery

I3-2 – Back-up Facilities

I3-3 – (indication suitability)

I3-4 – (Inhibit resetting of cumulative measurement values)

I3-5 – Dynamic behaviour

More detailed explanation of requirements and validation guidance are presented in attachment 10.1.

7.2.3 Expected results

The intention of the experiment was to check the following issues:

1. Identification of the requirements
2. Fulfilment of the requirements
3. Test methods used
4. Contents of the report
5. Text to be included in the Type Approval Certificate

Major hesitations regarded the equivalence of both identification of compulsory requirements and assessment of fulfilment of the requirements. Complete uniformity was not expected in the first iteration; the intention was rather to identify weak points in order to remove them.

7.3 Outcomes

By 5 p.m. on October 19th, MIRS had received test reports from NMi[5], MIRS[6], PTB[7], BEV[8], GUM[9] and CMI[10] (chronologically). In the summary report the institutions are named A, B, C, D, E and F. There is no relation between the time of individual test report arrival and the alphabetical denotation of the individual participant. The meeting of the Group for Analysis of the Results (GAR) took place at MIRS, Ljubljana, on October 23rd and 24th, 2006. The following are the results of the analysis:

7.3.1 Organisational and logistic issues

- The packages with equipment under test (EUT) prepared by Landis+Gyr were complete and enabled immediate start of the testing.
- The submitted documentation was complete.
- The organisation and management of the experiment were very good.
- The time frame of the experiment was appropriate for supporting the implementation of the MID.

7.3.2 Technical issues

At first glance the result seemed very dissatisfactory. Only nine requirements out of twenty-one have been judged as fulfilled by all the participating laboratories. The findings regarding the fulfilment of the remaining twelve requirements were different – and the differences were not the same. Extremely worrying was the fact that the overall conformity judgements were different: four participating laboratories found that EUT conformed to the MID, whereas the determination of two laboratories was that EUT did not conform. This caused an immediate, thorough analysis of the reasons, and preparation of corrective actions. A condensed overview of the outcomes of the experiment is presented in Table 7-1 below. Shaded cells indicate the requirements for which the outcome (pass/fail) is not equal with all participants, regardless of the test methods applied.

[5] Nederlands Meetinstituut (Netherlands Metrology Institute - http://nmi.nl/)

[6] Metrology Institute of the Republic of Slovenia (http://www.mirs.si)

[7] Physikalisch-Technische Bundesanstalt (German National Metrology Institute - http://www.ptb.de/)

[8] Bundesamt für Eich- und Vermessungswesen (Austrian National Metrology Institute - http://www.bev.gv.at/)

[9] Główny Urząd Miar (Polish National Metrology Institute - http://www.gum.gov.pl/pl/site/)

[10] Český metrologický institut (Czech Republic National Metrological Institute - http://www.cmi.cz/)

	A			B			C			D			E			F	
Requirement	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**
P1 – Documentation	+	D	+	+	D	-	+	D	+	+	D	+	+	D	+	+		+
P2 – Software identification	+	D,F	+	+	D,F	-	+	D,F	+	+	D,F	-	+	D,F	+	+		+
P3 – Influence via user interface	+	D,F	+	+	D,F	-	+	D,F	+	+	D,F	+	+	D,F	+	+		+
P4 – Influence via communication interface	+	D,F	+	+	D,F	-	+	D	+	+	D,F	+	+	D,F	+	+		+
P5 – Protection against accidental or unintentional changes	+	*D,F*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D,F*	+	+		+
P6 – Protection against intentional changes	+	*D,F*	+	+	*D*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+		+
P7 – Parameter protection	+	D	+	+	D,F	-	+	D,F	+	+	D,F	+	+	D	+	+		+
L1 – Completeness of measurement data stored	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D*	+	+		+
L2 – Protection against accidental or unintentional changes	+	D	+	+	D	+	+	D	+	+	D,F	-	+	D,F	+	+		+
L3 – Integrity of data	+	D	+	+	D,F	+	+	D	+	+	D,F	+	N/A		N/A	+		+
L4 – Authenticity of meas. data stored	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D*	+	+		+
L5 – Confidentiality of keys	N/A		N/A	N/A		N/A	N/A		N/A	N/A		N/A	N/A		N/A	N/A		N/A
L6 – Retrieval of stored data	+		+	+	D,F	+	+	D	+	+	D,F	-	+	D,F	+	N/A		N/A
L7 – Automatic storing	+	*D*	+	+	*D*	+	+	*D,F*	+	+	*F*	+	+	*D,F*	+	+		+
L8 – Storage capacity and continuity	+	*D*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+		+
I3-1 – Fault Recovery	+	D,F	+	+	D	?	+	*D*	+	+	D,F	+	+	D	+	+		+
I3-2 – Back-up Facilities	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D,F*	+	+	*D*	+	+		+
I3-3 –Wake-up facilities and restoring???	+	D	+	N/A		N/A	N/A		N/A	N/A		N/A	N/A		N/A	N/A		N/A
I3-3 – (indication suitability)	+	*D*	+	+	*D,F*	+	+	*D*	+	+	*D*	+	+	*D*	+	+		+
I3-4 – (Inhibit resetting of cumulative meas. values)	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+		+
I3-5 – Dynamic behaviour	N/A		N/A	N/A		N/A	+	D	+	N/A		N/A	N/A		N/A	N/A		N/A

Table 7-1: Overview of the outcomes: IDE: Requirement Identified, MET: Methods Used, PAS: Pass (+)/Fail (-), D: Documentation Analysis, F: Functional Test, N/A Not Applicable

7.3.2.1 Identification of requirements

Regardless of the qualifications of the staff, excellence of the methods and the available instruments and tools – exactness of the requirements, is the first prerequisite for a testing process. Every testing process begins with clarification of the requirements against which the testing is performed. Having in mind the intended application of the validated WELMEC Guide 7.2, the obvious precondition for an equivalent testing process was that every testing laboratory designated identical testing requirements. An overview of the identified requirements is presented in Table 7-2 below.

Requirement	A	B	C	D	E	F
P1 – Documentation	+	+	+	+	+	+
P2 – Software identification	+	+	+	+	+	+
P3 – Influence via user interface	+	+	+	+	+	+
P4 – Influence via communication interface	+	+	+	+	+	+
P5 – Protection against accidental or unintentional changes	+	+	+	+	+	+
P6 – Protection against intentional changes	+	+	+	+	+	+
P7 – Parameter protection	+	+	+	+	+	+
L1 – Completeness of measurement data stored	+	+	+	+	+	+
L2 – Protection against accidental or unintentional changes	+	+	+	+	+	+
L3 – Integrity of data	+	+	+	+	+	+
L4 – Authenticity of measurement data stored	+	+	+	+	+	+
L5 – Confidentiality of keys	N/A	N/A	N/A	N/A	N/A	N/A
L6 – Retrieval of stored data	+	+	+	+	+	-
L7 – Automatic storing	+	+	+	+	+	+
L8 – Storage capacity and continuity	+	+	+	+	+	+
I3-1 – Fault Recovery	+	+	+	+	+	+
I3-2 – Back-up Facilities	+	+	+	+	+	+
I3-3 – Wake-up facilities and restoring	+	-	-	-	-	-
I3-3 – indication suitability	+	+	+	+	+	+
I3-4 – Inhibit resetting of cumulative measurement values	+	+	+	+	+	+
I3-5 – Dynamic behaviour	N/A	N/A	+	N/A	N/A	N/A

Table 7-2: Designation of the compulsory requirements

N/A:Not Applicable
+: The requirement was identified as compulsory
-: The requirement was not identified as compulsory

Shaded cells indicate the requirements the testing of which has been designated differently among the participants of the experiment. In this case study 3 requirements out of 21 were interpreted differently, as presented in Table 7-3 below:

Requirement	Cause of different identification
I3-5 – Dynamic behaviour: The non-legally relevant software shall not adversely influence the dynamic behaviour of a measuring process.	Different level of detail analysis – participant C had analysed a wider background of the requirement
I3-3 – Wake-up facilities and restoring	Wrong naming of the requirement
L6 – Retrieval of stored data The software used for verifying measurement data sets stored shall display or print the data, check the data for changes, and warn if a change has occurred. Data that are detected as having been corrupted must not be used.	Misunderstanding of the meaning of the requirement (understood as verification of the software instead of verification of stored data sets).

Table 7-3: Causes for different designation of the requirements

The following reasons for different designations of the requirements were determined:

- deficient documentation of the unit under test,
- improper explanation in the guidance document,
- different level of detail analysis by testing engineers, and
- lack of language knowledge (latent danger of misunderstanding the texts written in a foreign language must not be neglected).

7.3.2.2 Applied test methods

All participants have used the same methods in testing of only 3 out of 21 requirements. It appeared, however, that the applied methods did not directly influence the results. The requirements that have not been tested using the same methods by all participants are indicated as shaded rows in Table 7-4 below.

Requirement	A	B	C	D	E	F
P1 – Documentation	D	D	D	D	D	/
P2 – Software identification	D,F	D,F	D,F	D,F	D,F	/
P3 – Influence via user interface	D,F	D,F	D,F	D,F	D,F	/
P4 – Influence via communication interface	D,F	D,F	D	D,F	D,F	/
P5 – Protection against accidental or unintentional changes	D,F	D	D,F	D,F	D,F	/
P6 – Protection against intentional changes	D,F	D	D	D,F	D,F	/
P7 – Parameter protection	D	D,F	D,F	D,F	D	/
L1 – Completeness of measurement data stored	D	D,F	D,F	D	D	/
L2 – Protection against accidental or unintentional changes	D	D	D	D,F	D,F	/
L3 – Integrity of data	D	D,F	D	D,F	N/R	/
L4 – Authenticity of measurement data stored	D	D,F	D,F	D	D	/
L5 – Confidentiality of keys	D	N/R	N/R	N/R	N/R	/
L6 – Retrieval of stored data	D	D,F	D	D,F	D,F	/
L7 – Automatic storing	D	D	D,F	F	D,F	/
L8 – Storage capacity and continuity	D	D	D,F	D,F	D	/
I31 – Fault Recovery	D,F	D	D	D,F	D	/
I3-2 – Back-up Facilities	D,F	D,F	D	D,F	D	/
I3-3 – Wake-up facilities and restoring???	D	N/R	N/R	N/R	N/R	/
I3-3 – Indication suitability	D	D,F	D	D	D	/
I3-4 – Inhibit resetting of cumulative Measurement. Values)	D,F	D,F	D,F	D,F	D,F	/
I3-5 – Dynamic behaviour	N/R	N/R	D	N/R	N/R	/

Table 7-4: Applied test methods

D –Analysis of the documentation
F –Functional Test
N/R. –Not Relevant
/ - No data

7.3.2.3 Fulfilment of the requirements

Twelve requirements out of twenty-one have not been met, have been interpreted differently, or have been met with a significant comment. An overview is presented in the following Table 7-5. Shaded cells indicate requirements for which the judgement of conformity was not the same among the participants of the exercise.

Requirement	A	B	C	D	E	F
P1 – Documentation	+		+	+	+	+
P2 – Software identification	+	-	+	-	+	+
P3 – Influence via user interface	+	-	+	+	+	+
P4 – Influence via communication interface	+	-	+	+	+	+
P5 – Protection against accidental or unintentional changes	+	+	+	+	+	+
P6 - Protection against intentional changes	+	+	+	+	+	+
P7 – Parameter protection	+	-	+	+	+	+
L1 – Completeness of measurement data stored	+	+	+	+	+	+
L2 – Protection against accidental or unintentional changes	+	+	+	-	+	+
L3 – Integrity of data	+	+	+	+	N/A	+
L4 – Authenticity of measurement data stored	+	+	+	+	+	+
L5 – Confidentiality of keys	N/A	N/A	N/A	N/A	N/A	N/A
L6 – Retrieval of stored data	+	+	+	-	+	N/A
L7 – Automatic storing	+	+	+	+	+	+
L8 – Storage capacity and continuity	+	+	+	+	+	+
I3-1 – Fault Recovery	+	?	+	+	+	+
I3-2 – Back-up Facilities	+	+	+	+	+	+
I3-3 – Wake-up facilities and restoring???	+	N/R	N/R	N/R	N/R	N/R
I3-3 – (indication suitability)	+	+	+	+	+	+
I3-4 – (Inhibit resetting of cumulative Measurement. Values)	+	+	+	+	+	+
I3-5 – Dynamic behaviour	N/A	N/A	+	N/A	N/A	N/A

Table 7-5: Fulfilment of the requirements: Overview by participants

N/A:	Not Applicable
N/R:	Not Relevant
+:	The requirement has been fulfilled
-:	The requirement has not been fulfilled
?:	Requires additional explanation

The reasons for non-conformance judgements in the test reports of the participants in the exercise are presented in Table 7-6 below. Additional comments are presented in Table 7-7.

Requirement	**P**	**F**	**NA**	**Reason for Fail or *Comment***
P1 – Documentation	5	1	-	B: Missing documentation of the watchdog clock *C: Partially ok, software identification performed by Firmware ID, no information about the procedure for the calculation of this ID was found.*
P2 – Software identification	4	2	-	B: The software ID could be read-out by the submitted software tools (F), however, it could not be indicated on the display D: The software ID is not automatically generated (checksum). See acceptable solution P2. *C: Partially ok, software identification performed by Firmware ID, no information about the procedure for the calculation of this ID was found.*
P3 – Influence via user interface	**5**	**1**	-	**B: If the tools are used, the utility seal is broken, the verification seal is not broken, and then the security level 3 is reached (see (E5) Section 16). At this level it is possible for the user to write to some registers that are crucial, e.g. the energy registers can be reset. This access that doesn't require destroying the verification seal is not allowed according to MID Annex I Paragraph 8.5. This is a reason for having failed the software validation.**
P4 – Influence via communication interface	**5**	**1**	-	**B: Write access to crucial registers is possible by communication via any of the interfaces of the instrument. Therefore deficiencies described in P3.3 and P7 also lead to failing requirement P4.** *D: What about interfaces sealed at installation under the connection cover (S0/CS, M-Bus)? Should the seal mentioned in the text be defined more precisely, as a manufacturer seal or seal applied by the »notified body«/«legal metrology institution«?*
P7 – Parameter protection	**5**	**1**	-	**B: Security level 3 gives too many permissions to the utility.**
L2 – Protection against accidental or unintentional changes	5	1	-	D: An error message is shown on the display, however, according to the documentation (13.3.3.) it is not clear whether the corrupted data can be read out normally or not. *E: Warning prior deletion or changing the parameters. It was accepted that password requirement is a warning to the user that he is accessing a level where he can change a parameter important to the proper functioning of the instrument. Since the same procedure is done at the software connecting to the instrument, there is a question whether there is need to define the requirement for the connection software to also warn the user about the possible consequence of the action each time the user tries to change the data.* *In this way the legal control does not end at the instrument but it expands the requirement of warning not only to the user interface at the instrument itself but also to the indirect/remote user interface (remote user interface in the software).*

[5] ZMD110AR – ZMD120AR Functional Description

Requirement	**P**	**F**	**NA**	**Reason for Fail or *Comment***
L6 – Retrieval of stored data	4	1	1	D: An error message is shown on the display, however, according to the documentation (13.3.3.) it is not clear whether the corrupted data can be read out normally or not.
I3-1 – Fault Recovery	5	?	-	B: There are three kinds of error flags that indicate wrong programme execution ((E) Section 13.3.4), however, it is not documented whether the software of the instrument is able to recover, when it is in an undefined state, e.g. after an electromagnetic disturbance. In such a situation it could be impossible for the programme to set one of the error flags. An appropriate means would be a watchdog that resets the micro-controller even in an undefined state. A watchdog is mentioned in (E) 13.3.4, but its function is not described in detail. Additional documentation is necessary.
I3-5 – Dynamic behaviour	1	-	5	C: The operating kernel has a high priority queue for measuring system messages and a second one for lower priority tasks. Therefore the regular processing of measurement values is ensured (“MID-Software description ZMD100AR”, Chapter 4.1). Interrupt hierarchy and timing diagram of the different software tasks are missing in the documentation.

Table 7-6: The reasons for non-conformance judgements

Comments				
L4 – Authenticity of measurement data stored				E: Functional check not possible F: Only load profile was considered in connection with this requirement.
L3 – Integrity of data				F: Only load profile was considered in connection with this requirement.
P5 – Protection against accidental or unintentional changes				D: Protection is by hardware seal. C: Partially ok, LR data are protected by Checksum, no statement about protection of SW was found?? (See “Functional Description”, Chapter 13.3.).
P6 – Protection against intentional changes				D: Protection is by hardware seal.
P7 – Parameter protection				D: Protection is by hardware seal. C: Comment: it has to be stated in the type approval document (conformity assessment document), which data are under legal control and can be used for billing (e.g. data provided by pulse outputs).
L8 – Storage capacity and continuity				D: The storage device satisfies the requirement L8. However, it does not comply with specifying note 1, which states that a warning shall be given when the storage is full. In this case this note is not applicable.
I35 – Dynamic behaviour				D: No software separation.
I3-2 – Back-up Facilities				D: The meter satisfies the requirement. However, the specifying note requires that the back-up period is such that the critical change value is not exceeded. The term critical change value needs further explanation.

Table 7-7: Additional coments to judgements

Other remarks:

1. It is not clear which parts of the instrument should be considered as legally relevant. Participant “D” means that the cumulative energy register is the preliminary observation which is used for trade. Therefore, only the cumulative energy register (and the fault recovery back-up) is legally relevant, and additional data logging devices are not.

   However, it is not clear whether this opinion is shared by the surveying authorities throughout Europe. This could cause problems during inspections and re-verifications. A united point of view is necessary.

   Also the preliminary observation can be different for each type of instrument under the MID. Therefore, the extent of the preliminary observation for each type of instrument must be stated in Extension I to the Guide.
2. Article 10.3.5 of the MID states that cumulating energy register of domestic instruments is not a long-term storage in the sense of Extension L. What is the exact definition of the term “domestic”? This also means that it depends on the location of the instrument. It might be necessary to mention in the certificate that the instrument can only be used for domestic applications if the cumulative energy register is not examined according to Extension L.
3. The specifying notes of requirement L8 state that a warning must be issued when the storage is full, regardless of the intended purpose. In this case there is a storage after each billing period (e.g. each month). The storage has capacity for multiple billing periods (e.g. 10 months). Overwriting the monthly storage after 10 months is not a problem. A warning is therefore not required.
4. The specifying note of requirement I3-2 states that the minimum interval of back-up has to be calculated to ensure that the critical change value is not exceeded. It is not clear what the definition of “critical change value” is.

7.3.2.4 Summary of the reasons for different judgements of non-conformity and consequent necessary improvements

7.3.2.4.1 Insufficient functionality of the unit under test and the necessary amendments

- According to the essential requirements of MID (8.5), registers that store data about cumulative consumed energy shall not be reset during use. The discrepancy of the unit under test submitted to comparative validation was because writing to these register was possible without destroying the verification seal, within access level 3.. The manufacturer needed to reorganise the access rights in such a way that, e.g., writing to these registers was enabled only at access level 4. This change resulted in fulfilment of the requirements P3, P4 and P7.
- According to the essential requirements of MID (8.3), software that is critical for metrological characteristics shall be identified as such and shall be secured. Software identification shall be

easily provided by the measuring instrument. In the case of the unit under test submitted to comparative validation it was possible to configure the meter in such a way as to provide software identification via remote interface (e.g. infrared), which, according to the opinion of the examiners, is not "easily provided". It was necessary to amend the functionality of the displaying software identification on demand given from the meter front panel buttons (without additional tools such as PC and software). This change resulted in the fulfilment of the requirement P2.

7.3.2.4.2 Insufficient documentation of the unit under test

Several discrepancies were caused by insufficient documentation: P1, L6, I3-1, and I3-5. Improvements of the documentation resulted in the fulfilment of all these requirements.

7.3.2.4.3 Necessary amendments of the Guide

Several points in the Guide allowed different interpretations of either the requirements or technical solution for their fulfilment, e.g.:

- It turned out that it was not clear what was meant under "user interface":
  - only the buttons operated by human fingers and the display visible by human eye, or
  - a user interface using a portable PC, special software and communication interface (e.g. communication cable).
- The explanation of "easily presented" was ambiguous; e.g.:
  - is it on demand by pressing the button on the front panel, or
  - it requires a series of actions (connection to a portable PC, special software and communication interface, entering password, setting the parameters in the meter, sending a command and presentation of e.g. software identification on the PC display).

7.3.2.5 Problems beyond the scope of the WELMEC Guide 7.2

Although in principle harmonised within the European legal metrology area, some country-specific requirements from legislation other than metrological influence the measuring instrument's functionality.

Examples of such requirements are:

- Legal relevance of the load profile registers. In some countries only the registers storing cumulative energy consumption are legally relevant; in some other countries load profile registers are legally relevant as well.
- Sufficiency of long-term storage capacity. This requirement is influenced by taxation legislation. For the requirement that long-term storage "must have a capacity which is sufficient for the intended purpose" – data retention period for measured values may be, depending on the country, from one to more than five years – which directly affects the design of the device (size of memory).

Such variations of requirements may cause problems to legal metrology institutions in implementing harmonised conformity assessments – the result may be, e.g., a limited or conditional type approval certificate for a measuring instrument. On the other hand, this is just one example of real-life situation – it has been anticipated that complete harmonisation will not be possible and that some minor departures will remain.

7.3.3 Issues related to documentation connected with the type evaluation process

7.3.3.1 Test plan

Understanding of the necessary contents and form of the test plan was different among the participants of the exercise. Since this could lead to a different extent of testing, the definition of the sample test plan in the WELMEC Guide 7.2 should be re-considered.

7.3.3.2 Test report template

The test report template proposed in the WELMEC Guide 7.2 has been approved as adequate by all participants in the experiment.

7.3.3.3 Text to be included in the Type Approval Certificate

One of the required outputs of the validation was a section of text intended for subsequent use by field inspectors, which provides essential information on the measuring instrument's software, needed for the inspection of measuring instruments in use. This section should comprise the following information:

- Reference to the documentation submitted for type approval;
- Identification and description of the electronic (hardware) components (subassemblies, modules) that are important for software/IT function of the measuring instruments;
- Overview of the software environment, which is necessary to operate the software;
- Overview of SW modules under legal control (including SW separation, if implemented);
- Overview and identification of hardware and software (if relevant) interfaces that are important for software / IT functions of the measuring instruments (including infrared, Bluetooth, Wireless LAN, …);
- Identification and description of locations of software components in the measuring; instrument (i.e. EPROM, processor, hard disk, …) that need to be sealed or secured;
- Instructions on how to check the identification of software (for metrological supervision);
- In the case of electronic sealing: instruction for the inspection of audit trails.

Participants in the comparative software validation had different approaches regarding both the extent and contents of this section:

- No text (A, C);
- Reference to the documentation where relevant information can be found (B);
- Description of the SW identification only (D);

- Very short description (F);
- Very comprehensive (5 pages) description including pictures and description of HW (E).

This obviously was not a high priority issue for the performers of the validation. Nevertheless, the presetting of this information in an optimised form is very important for inspection purposes during instrument's use, and has to be considered in the future.

7.3.3.4 Guidance on how to prepare the documentation for applicants

It proved during the experiment that well-prepared documentation of the unit under test is extremely important. In this case study there was no need for additional manufacturers' explanations about the evaluated software, which enabled the conclusion of the validation in two months' time, effectively.
Two documents emerged as key guidance for analysing the documentation of the unit under test:

- The document "MID – Software declaration ZMD100AR" (which constituted a part of the submitted documentation) was very useful for the performers. Since it contains explanations as to where answers to particular requirements are described, it saved a significant amount of time during the analysis of the documentation. It was strongly recommend by GAR that such a document become a part of the documentation submitted to each future type approval.
- A table with classification of legal relevance of parameters should be a part of the type approval documentation (an example for electricity meters is proposed in Table 10-13 in Attachment 10.7).

7.4 Analysis of success of the experiment

From the application-specific point of view the outcomes of the experiment may be summarized as follows:

7.4.1 Justification of the experiment

According to first analysis of the experiment of comparative software validation exercise, two thirds of participating laboratories confirmed inappropriate measuring instrument's software. In other words, had this experiment and the necessary follow-up corrective actions not been performed, the danger of a measuring instrument with improper software passing conformity assessment would have been with 66.7% probability.
This fact confirms the necessity for the performance of such an experiment. Figure 7-1 illustrates the participants' findings on fulfilment of each particular requirement, while the Figure 7-2 illustrates the findings on fulfilment of requirements of each particular participant.

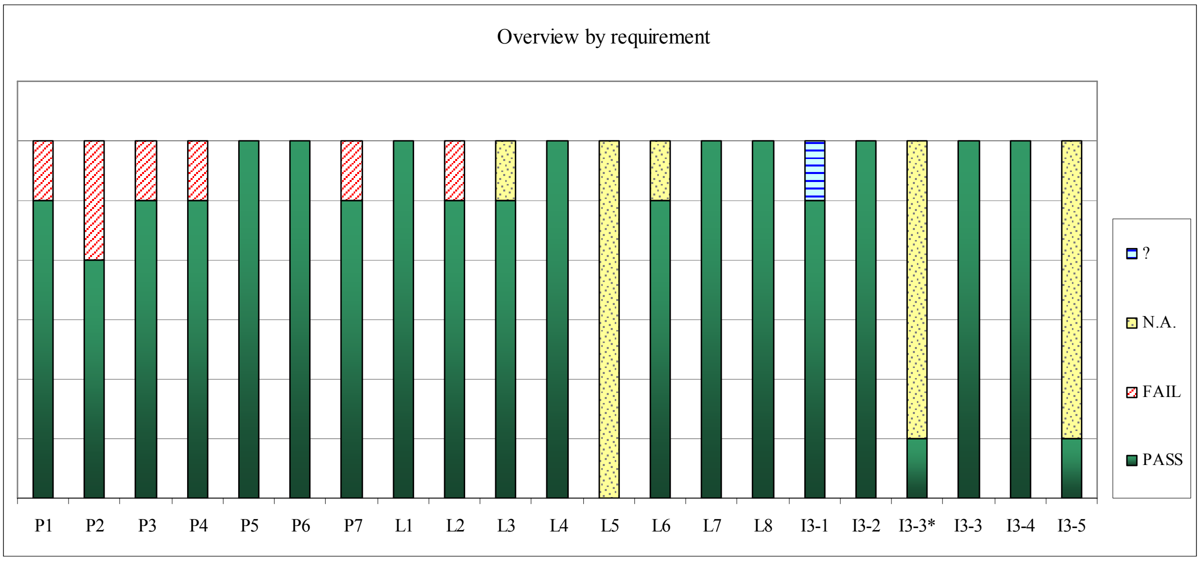


Figure 7-1: Overview of fulfilling the requirements, by requirement

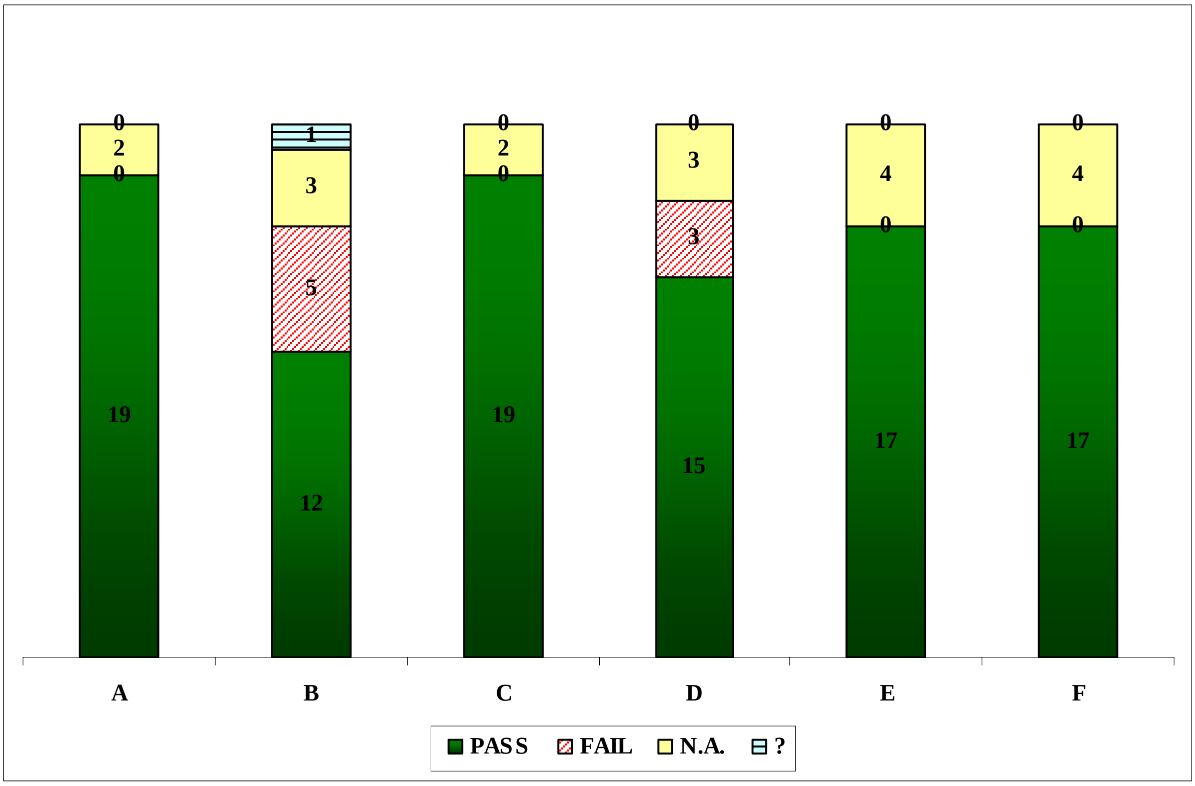


Figure 7-2: Overview of fulfilling the requirements, by participants

Two issues that are not easily determinable and are hard to control by technical means, are related to the individual characteristics of a particular examiner – the strictness and depth of detail analysis. The consequences of deviations arising from such personal attributes can be effectively compensated by a feedback loop in the form of the described comparative validation. Only the final results of validation of individual laboratories were observed and compared in the report on the comparative validation experiment, and in this thesis – an analysis of the causes for the discrepancies is left to each individual laboratory itself.

7.4.2 Participating laboratories sample

Another thing that must be taken into consideration is the quality of the sample of participating laboratories. Ideal conditions for the performance of such an exercise would be:

- existence of a wide base of competent laboratories, and
- random selection of participants.

Comparing to these ideal conditions, there were two facts countering the selection of laboratories in this experiment:

- all participating laboratories volunteered for participating in this exercise,
- experts from all participating laboratories were involved in the process of preparation of the WELMEC Guide 7.2, and could therefore have been influenced by the background information they obtained during the preparation of the Guide.

The latter deficiency could not be avoided in this experiment, because there were no other competent software examination laboratories in the European legal metrology community at the time of the experiment. Repetition of the experiment at some later date, when more laboratories become skilled for software validation, would be of great benefit.

7.4.3 Analysis of the performance of testing

In this sub-section an analysis of the relations between the applied testing methods and the obtained results among individual participants is presented.

The assumed values are as follows:

For estimation of the invested effort; it was assumed that performing a functional test was more time-consuming than analysing the documentation. This estimation is very rough, based on deficient information about the test methods used:

- analysis of the documentation: weighting factor 1,
- performance of functional tests: weighting factor 2.

The result of testing individual requirement:

- Requirement not met: -1;
- Has not been tested: 0;
- Requirement met: 1;
- Test OK: the result of testing the particular requirement is equal to the reference result;,
- Test not OK: the result of testing the particular requirement is not equal to the reference result;,
- Reference test result: an agreed upon correct value of the test result (most stringent result, if applicable).

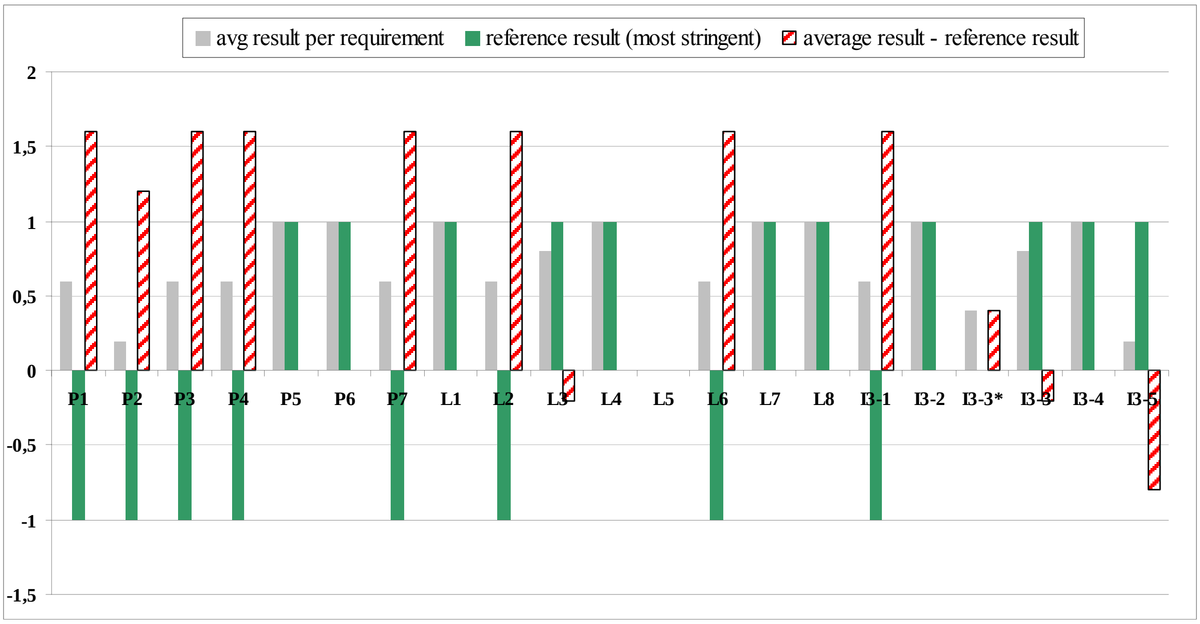


Figure 7-3: Average result, reference result and deviation of average result from the reference result per particular requirement

The data presented in Figure 7-3 best illustrate the understanding individual requirements. Taking into consideration the difference between the reference result and the average result as criteria, it is evident that the best understood requirements were the following: P5, P6, L1, L4, L5, L7, L8, I3-2 and I3-4. The worst understood requirements were: P1, P3, P4, P7, L2, L6 and I3-1. The conclusion of this part is that the requirements from the second group need better explanation in the WELMEC Guide 7.2.

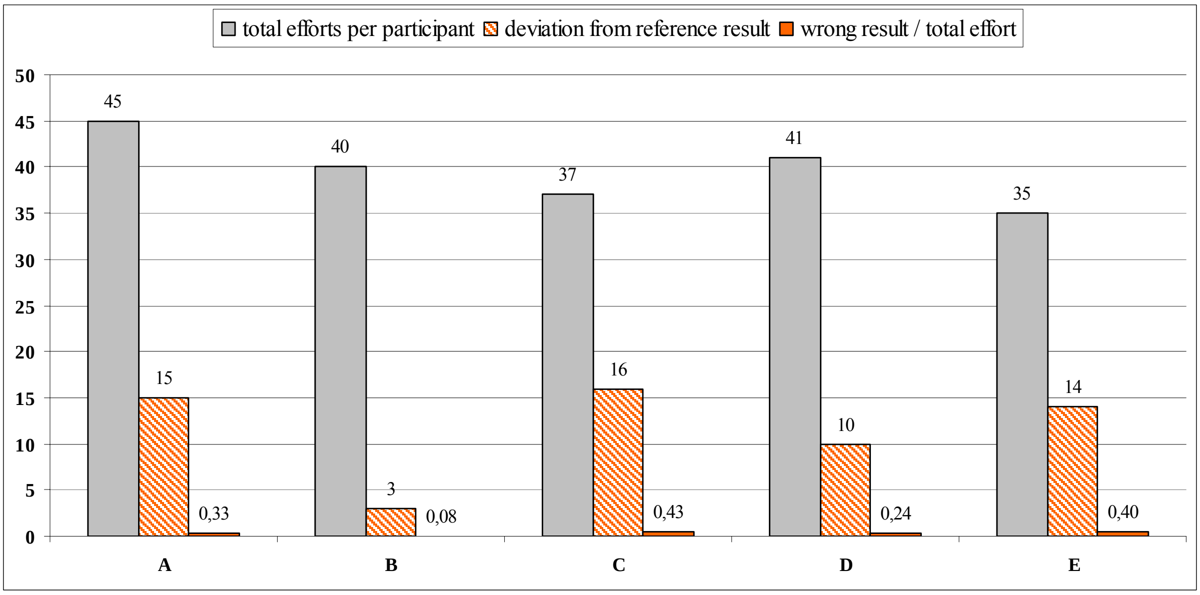


Figure 7-4: Total efforts, deviation from reference result and efforts per wrong results per participant

The data presented in Figure 7-4 provide direct information about the testing skill of a particular participant. For example, the deviation of the result of participant "B" from the reference result was only 3, while the deviation of participant "C" was 16. Another parameter illustrates efforts invested in wrong judgements: while participant "B" had only 0.08 wrong judgements per one unit of effort, participant "C" had 0.43 bad judgements per same unit of effort. By simplifying this estimation, one may say that participant "C" made 5,375 times more wrong judgements than participant "B". This is a significant parameter, which enables a laboratory to improve its work.

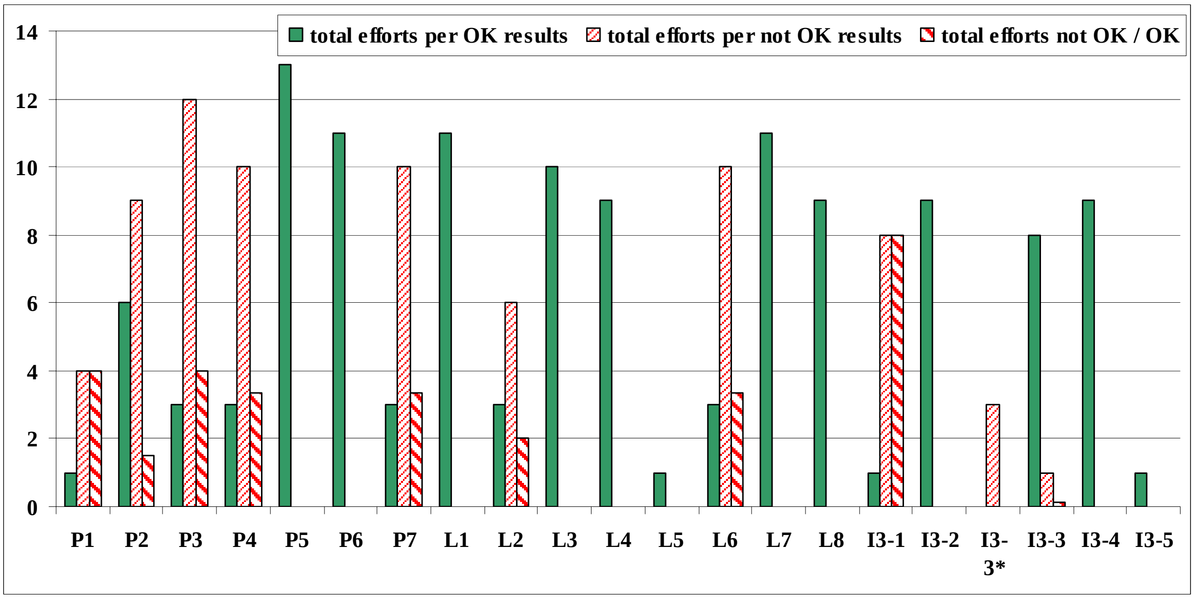


Figure 7-5: Summary of efforts per correct and wrong judgements per requirement

Figure 7-5 provides the information on which requirement most effort had been invested for the achievement of bad results. Similarly as described in Figure 7-3, these requirements are: I3-1, P1, P3, P4, P7, L6, L2 and P2.

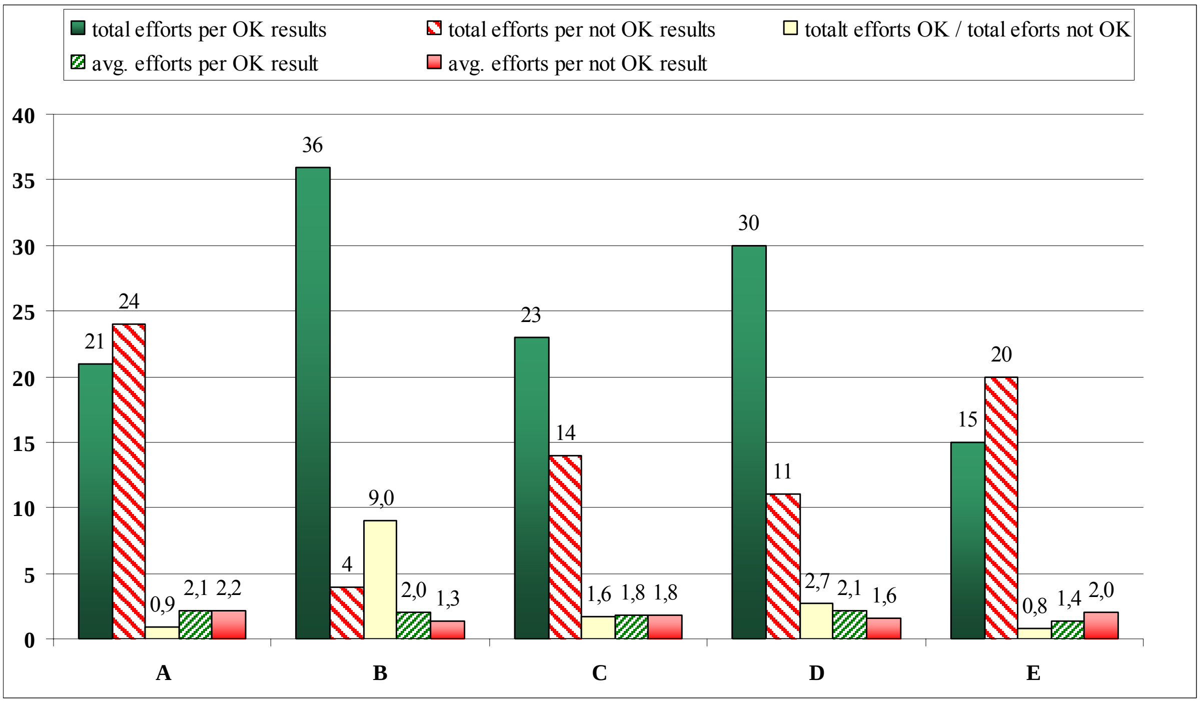


Figure 7-6: Efforts per correct and wrong results per participant

Figure 7-6 gives another view of the efficiencies of individual laboratories, highlighting the distribution of work between efforts that led to wrong judgement and efforts giving adequate results. The parameter "total efforts OK / total efforts not OK" gives an estimation of the efficiency of work of a particular laboratory. The value of this parameter describes the ratio of efforts invested in the achievement of a correct judgement, and efforts invested in the achievement of a wrong judgement. Simplified, one may say that laboratory "B" is 11.25 times more efficient than laboratory "E". Although it was not the original intention of the exercise, it became obvious that

- The amount of work does not directly influence the quality of a testing process.
- The selection of methods does not necessarily influence the quality of a testing process.

The main influences on the quality of a testing process are

- proper understanding of the requirements, and
- performer' skill.

7.5 Conclusions, lessons learned, next steps

Comparative measurements or testing of the same artefact between different performers (laboratories) is an already established practice in many areas, from inter-comparisons at the highest level devoted to maintenance of worldwide quality of measurements in particular metrological areas (key comparisons) [1], followed by comparison of measurements of the same artefact performed by different testing or calibration laboratories (enabling improvement of their measurement capabilities), to proficiency testing, which may even be understood as comparison and ranging of laboratories' measurement capabilities. Certain guidelines for the performance of proficiency testing already exist, e.g. ISO/IEC Guides 43-1 and 43-2 [25], and are intended mostly for material testing and chemical metrology laboratories.

Comparative software validation in the area of validation of metrological software was performed for the first time, so no systematic knowledge on this topic existed at the time of beginning the experiment. The aim of the experiment presented herein was not proficiency testing, but rather validation of the guidance document and acquiring knowledge in possibilities of harmonised implementation of existing software validation approaches in the specific field of validation of metrological software. Nevertheless, the analysis of the results revealed some parameters that could be very useful for assessing the quality of a laboratory's work, as well as for its improvement.

Comparative validation of a software component has one significant advantage over other comparison examinations; there is no need for the unit under test to be transported between participants. Certainly, each participant needs to have one sample – a measuring instrument with embedded software under test, but in this case it is possible to provide the required number of samples with the same software for all participants simultaneously – it is not necessary that all participants should perform the testing on exactly the same sample. Consequently, all participating laboratories may perform the work simultaneously, and the duration of the experiment is significantly shorter.

7.5.1 Outcomes of the experiment

The validation of the WELEMC Guide 7.2 by comparative validation proved its suitability for the intended purpose – to enable comparable, harmonised approach to conformity assessment of software embedded in the measuring instruments covered by the EU Measuring Instruments Directive. Some findings were made during the validation, which have already been used for the improvement of the Guide (WELEMC Guide 7.2 Issue 2), and are significant to the common understanding of legal metrology in general.

Performance of similar experiments in the future will be of great advantage for the improvement of the software validation procedures. The presented inter-comparison is related to the simplest software configuration of the instrument and to the conformity assessment module B (“type examination”). It will be of great benefit to validate the parts of the Guide related to a more complex IT configuration of measuring instruments (e.g. based on a universal computer, an instrument with transmission of data via IT networks, or with envisaged remote software update). Future development will very probably involve an expansion of the Guide to other conformity assessment modules (especially H1: “Declaration of conformity based on full quality assurance plus design examination”, which is of great interest for manufacturers of measuring instruments). Inter-comparison of approaches to H1 conformity assessment module will be very important for the legal metrology community. Nevertheless, comparative validation may provide a very effective tool for the improvement or maintenance of the already achieved level of equivalence in many other domains, not only in software validation in legal metrology.

7.5.2 Necessary improvements of the Guide

- P2 (Basic Requirements for Embedded Software in a Built-for-purpose Measuring Instrument): "SW identification for verification and inspection purposes has to be easily shown". The term "easily" is not a proper technical definition, it is not clear whether it means "by standard user interface only", e.g. front panel buttons, or additional tools as well (e.g. a readout unit, PC software and communication cable);
- P2 – SW identification: Check sum is just one of the acceptable solutions, it is not required. This misunderstanding happens very often in the interpretation of technical standards.
- P3, P4 and P7: There is a need for better definition of the user and communication interfaces. Proposals were the following:
    - User interface: To stress that it is the interface forming part of the instrument.
    - Communication interface: To stress that it is the interface that enables information to be automatically passed between the components of measuring instruments or sub-assemblies.

- Specifying notes for the requirement L8 ("Long time storage"): To be revised in the way that warning is not necessary if it is ensured by the design that only the overwriting of outdated data can happen;
- Requirement I3-2 (Specific Electricity Meter Requirements):The requirement needs clarification regarding acceptable behaviour in the case of the watchdog event and connection with the "critical change value".

Once again it has been proven that well prepared documentation is essential for the speed and quality of the examination process. Well considered documentation may spare both the applicant and the examination laboratory many iterations of communication, which could be very time-consuming. Document "H 71 0015 0029 en - ZMD100AR - MID Software Declaration" is an excellent example of well prepared documentation.

7.5.3 Conclusions

The proposed solution of development of validation guidance for metrological software has proven its suitability in practice. Experimental confirmation of usability of the guidance document detected several weaknesses, which were not hard to correct.

A second major achievement is the method for validation of suitability of such guidance documents, which was developed and tested in a practical experiment. The developed method of comparative validation of metrological software may be used in other areas as well.

8 Outline of expected contributions to science

The most important work explained in the present thesis comprises:

- Comprehensive analysis and evaluation of existing guidance documents with regard to their applicability to measuring instruments in legal metrology;
- Elaboration of the guidance document for validation of software in the measuring instruments covered by the Directive on measuring instruments. Since it had been ascertained that existing normative and guidance documents were not appropriate for the assigned task (validation of software in the measuring instruments covered by MID), it was necessary to develop a new guidance document;
- Confirmation that the developed guidance document is consistent with all relevant related fields besides that of software validation for legal measuring instruments. This is especially important in circumstances where manifold spheres are involved (technical: metrology, software quality, software testing, quality assurance; and social: legislative, conformity assessment, consumer protection, free movement of goods and services) – it is particularly important that the proposed new practice does not conflict with the established rules of all related spheres;
- Development and experimental operation of comparative validation of the same metrological software component between various performers as a method for validation of the developed guidance document.

The achieved contributions to science were the following:

1. **Extracting the relevant requirements, approaches, procedures and techniques from existing international software quality standards, software testing standards, normative documents related to metrological software and worldwide guidance documents related to metrological software**

This analysis was important for proving that the developed guidance document was in line with the approaches of technical domains of software quality and software testing practices. Relating these two groups of requirements ensured that the developed guidance for domain-oriented software validation fitted the essential requirements of software quality and testing. The performed work proved the existence of a straightforward relation.

The following identified weak points additionally confirmed this methodology:

- Insufficient coverage of the quality characteristic Usability (ISO/IEC 9126-1);
- Insufficient attention paid to the quality of user documentation. It is only mentioned as a necessary part of the documentation;
- WELMEC Guide 7.2 addresses security to a much greater extent than ISO/IEC 12119, especially the functional testing of security;

- Potential software failures are treated from the point of view of functionality (security) rather than reliability.

An overview of the available guidance documents on the quality and validation of metrological software revealed that all the documents arose spontaneously, when the particular group of users faced a problem that needed some guidance or standardisation. The documents cover software in measuring instruments or systems, laboratory administrative software, quality system management software as well as laboratory information systems. Some guides treat software as a product, others treat the software lifecycle processes. Target audience comprises manufacturers, conformity assessment experts and metrological experts, who develop measuring software applications by themselves.

Although the surveyed documents varied in the above-mentioned aspects, it was important to get an impression about current approaches in the metrological software validation community, in order to be able to confirm that the developed guidance was aligned with the state of the art in this particular technical field.

More details on these analyses are provided in Sections 6.1, 6.2 and 6.3.

2. **Establishing links between domain-specific standards for software components used in metrological applications, generic software quality standards, software testing standards, software testing methods and software testing strategies**

This subject is important as methodology for proving the connection between software validation techniques appropriate for a particular technical domain (e.g. software in measuring instruments) and the methods and strategies common to broad software testing practices. Relating these two groups of validation guidance makes it possible for the domain-oriented software validation approach to fit the essential requirements of software quality and testing. In addition, most applicable software testing approaches for non software testing experts were identified.

More details on this analysis are given in Section 6.5.

3. **Development of a generic procedure for software validation in measuring instruments and systems**

This particular issue is important because of the fact that there are several spheres in which software plays a significant role, and at the same time, the domain experts are not that familiar with the validation of software. The intention was to prove that it is possible to prepare an appropriate software validation guidance for the domain experts. The guidance document WELMEC Guide 7.2 is a result of group work. It is an example of successful application of this generic approach to the field of validation of measuring instruments' software in metrology, in particular to the software of instruments regulated by the Directive on Measuring Instruments.

It covers the complete process of validation of software in legal metrology measuring instruments (so called "embedded software") and systems, and is composed of the following phases:

- Identification of the essential software and IT-related requirements for particular groups of measuring instruments with regard to their technological realisation;
- Definition of the building block for modular structure;
- Refinement of these requirements to testable software requirements;
- Guidance for validation of these refined requirements for non-IT experts;
- Guidance for harmonised reporting of results and findings.

The guidance document in not applicable to the measuring instruments regulated by the Directive on Measuring Instruments only, but also to the validation of the software of measuring instruments used in non-harmonised and non-regulated areas.

Detailed explanation of the guidance document is available in Section 5.

4. Development and validation of the method, and comparative validation of a measuring instrument's software

The importance of this topic arises from the need to find the appropriate means for validation of a newly developed guidance (pre-standard) document for validation of metrological software. Such methodology (comparative validation of the same software component by several institutions) has never been used before in this technical field.

Performing an experiment was essential to prove the suitability of the proposed method for comparative metrological software validation. The analysis of the outcomes of the validation experiment gave interesting results, both for the improvement of the Guide and for the improvement of work of the laboratories who took part in the experiment.

At first glance there appeared to be important discrepancies between the identified conformities to particular requirements (fulfilment of only 9 out of 21 requirements matched with all participants). Further analysis, however, revealed that the main reason for these discrepancies lay in different understanding of particular requirements of WELMEC Guide 7.2. The necessary improvements of the Guide have already been implemented in Issue 3 of the Guide.

An interesting finding relates to the relationship between the applied test method and the outcome: there is no direct correlation. Consequently, there is no correlation between the effort invested in testing the particular requirement and the outcome, either. The skill of the performer of the testing plays the most important role. However, it is not possible to measure it directly.

The organisation of the experiment was appropriate, it enabled timely acquiring of the information about the possibility of harmonised assessment of software of the measuring instruments regulated by MID.

The developed methodology for comparative software validation offers at least three possible applications:

- Validation of newly developed guidance documents (standards);

- Determination of the degree of equivalence of approaches of individual laboratories to validation of metrological software (identification of issues to be improved – qualification of laboratories to participate in harmonised conformity assessment);
- A useful tool for maintaining the achieved degree of equivalence (periodical checking of the participating laboratories).

Although the main intention was not to rank the laboratories by quality of the performance, several findings point to that almost directly.

Detailed description of the experiment is available in Section 7.

9 Summary

Modern approach to conformity assessment puts accent on removing barriers to trade, promoting the so-called "one-stop testing" in order to minimize the related expenses and time to market for manufacturers. At the same time, it is of extreme importance to maintain a suitable level of consumer protection, as well as to ensure conditions for fair trade between new businesses in the area of metrology.

The only way to achieve this is to ensure an appropriate level of equivalence of conformity assessment between all institutions that perform conformity assessment. Such approach will have an additional economic impact, as it protects responsible and fair manufacturers of measuring instruments (those who do not apply unfair shortcuts for increasing their share on the market).

Software validation is the main conformity assessment activity in the field of measuring instruments' software. Harmonization in this area means ensuring a suitable level of equivalence of software validation among various performers.

Appropriate guidance is the prerequisite for harmonised validation of software in measuring instruments. Additionally, it is important for proper validation of software to be familiar with the specifics of metrological – in this case legal metrology –software applications (e.g. support for repeatability, reproducibility, protection of data, support for remote verification and inspection of instruments), as well as to be in-line with the state-of-the-art of its design (as presented in Section 2). These specifics are related to the specifics of measurement techniques, variety of technological implementations, reducing of operational costs, support for consumer protection, support for new business participants, support for remote legal verifications and inspection, compatibility with home-built software components, and supporting additional user functionality to improve market competitiveness.

Clearly, knowing the software testing techniques, as presented in Section 3, is inevitable.

At the time of preparation of the Directive on Measuring Instruments, several guidance documents for validation of metrological software existed (as explained in Section 4), but none of them was suitable for the validation of software in MID measuring instruments. This was a serious a problem for harmonised implementation of the Directive on Measuring Instruments.

This was also the background for the initiative to prepare a guidance document for validation of software in the measuring instruments covered by the MID (as described in Section 5). The initial research work for the core subject of the present thesis, namely the validation procedure for metrological software, was conducted during the development phase of the WELMEC Guide 7.2.

It was necessary to make sure that the new guidance document did not introduce any conflicting requirement with those of other normative documents or technical standards, as explained in Section 6. An interesting conclusion of this analysis is that most of the software validation can be done using very simple methods.

Next important step in ensuring the conditions for harmonised implementation of the Directive on Measuring Instruments in the field of metrological software was validation of applicability of the newly developed software validation guidance, WEMEC Guide 7.2. A comparative validation experiment of the same instance of software was chosen as the method. Six laboratories from the national metrology institutes of Austria, Czech Republic, Germany, Netherlands, Poland and Slovenia took part in the experiment. They all simultaneously performed validation of the software of an electric energy meter as part of type examination according to the "B" conformity assessment module.

Such an experiment had never been performed before in the field of metrological software, and two major achievements appeared after its conclusion. The first benefit of this experiment was the identification of weak points in Guide 7.2, which have already been resolved in the subsequent issues of the document. An additional new finding when analysing the outcome of the experiment were several parameters, which enabled the detection of weak points in the work of a testing laboratory. The following corrective actions led to improving the quality of laboratory work, and consequently, to better harmonisation of the validation approach and conformity assessment in general.

The performance of similar experiments in the future will be of great advantage to the improvement of software validation procedures. It would be of great benefit to validate parts of the Guide related to a more complex IT configuration of measuring instruments (e.g. based on universal computer, instrument with transmission of data via IT networks, or with foreseen remote software update). Future development will very probably include the expansion of the Guide to other conformity assessment modules (especially H1: “Declaration of conformity based on full quality assurance plus design examination”, which is of great interest for the manufacturers of measuring instruments). Inter-comparison of the approaches to H1 conformity assessment module will be very important for the legal metrology community. In addition, comparative validation may be a very effective tool for the improvement or maintenance of the already achieved level of equivalence between the performers of software validation in legal metrology.

It is important to point out that the presented work is not only applicable to legal metrology applications but also to other metrological fields.

Notes:

The web-enabled services presented herein are provided as examples only, and not as a complete list of metrological services available on the Internet. All web-sites referenced in this Section were active at the time of writing the thesis. There is no guarantee that they are still "alive", and not transferred to another web address.

Certain commercial entities, equipment, or materials, are mentioned herein in order to describe properly an experimental procedure or concept. Such identification is not intended to imply recommendation or endorsement, nor is it intended to imply that the entities, materials, or equipment are necessarily the best available for the purpose.

10 Attachments

10.1 Complete list of requirements and validation guidance for electricity meters

Requirement	Validation Guidance (for Risk Class C)
P1 – Documentation	
- A description of the legally relevant software. - A description of the accuracy of the measuring algorithms (e.g. price calculation and rounding algorithms). - A description of the user interface, menus and dialogues. - The unambiguous software identification. - 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. - The operating manual.	
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. 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.	Checks based on documentation: - Examine description of the generation and visualisation of the software identification - Check whether all programs performing legally relevant functions are clearly identified and described so that it is clear to both Notified Body and manufacturer which software functions are covered by the software identification and which are not. - Check whether a nominal value of the identification (version number or functional checksum) is supplied by the manufacturer. This must be quoted in the test certificate. Functional Checks: - The software identification can be visualised as described in the documentation. - The presented identification is correct. - The documentation (plus the executable code if necessary) of the pattern is kept at the NB.
P3 – Influence via user interface	
Commands entered via the user interface shall not inadmissibly influence the legally relevant software and measurement data. Required Documentation:	Checks based on documentation: - Judge whether all documented commands are admissible, i.e. whether they have an allowed impact on the measuring functions (and relevant data) or none at all. - Check whether the manufacturer has supplied an

If the instrument has the ability to receive commands, the documentation shall include: - A complete list of all commands (e.g. menu items) together with a declaration of completeness. - A brief description of their meaning and their effect on the functions and data of the measuring instrument.	explicit declaration of completeness of the command documentation. Functional Checks: - Carry out practical tests (spot checks) with both documented and undocumented commands. Test all menu items, if any.
P4 – Influence via communication interface	
Commands input via communication interfaces of the instrument shall not inadmissibly influence the legally relevant software and measurement data. Required Documentation: If the instrument has an interface, the documentation shall include: - A complete list of all commands together with a declaration of completeness. - A brief description of their meaning and their effect on the functions and data of the measuring instrument.	Checks based on documentation: - Judge whether all documented commands are admissible, i.e. whether they have an allowed impact on the measuring functions (and relevant data) or none at all. - Check whether the manufacturer has given an explicit declaration of completeness of the command documentation. Functional checks: - Carry out practical tests (spot checks), using peripheral equipment, if available Note: If it is not possible to exclude inadmissible effects on the measurement functions (or relevant data) via the interface, and the software cannot be amended accordingly, then the test certificate must indicate that the interface is non-protective, and describe the securing/sealing means required. This also applies to interfaces that are not described in the documentation.
P5 – Protection against accidental or unintentional changes	
Legally relevant software and measurement data shall be protected against accidental or unintentional changes. Required Documentation: The documentation should show the measures that have been taken to protect the software and data against unintentional changes.	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 foreseen and documented by the manufacturer. - Check that a warning is given to the user, when he is about to delete measurement data files. Functional checks: - Check by practical spot checks that a warning is given before deleting measurement data, when deleting is possible at all.
P6 – Protection against intentional changes	
Legally relevant software shall be secured against the inadmissible modification, loading or swapping of hardware memory. Required Documentation: The documentation shall provide assurance that legally relevant software cannot be inadmissibly modified.	Checks based on documentation: - Examine whether the documented means of securing against unauthorised exchange of the memory that contains the software are sufficient. - If the memory can be programmed in-circuit (without dismounting), check whether the programming mode can be disabled electrically and the means for disabling can be secured/sealed. (For checking

	download facilities see Extension D) Functional checks - Test practically the programming mode and check whether disabling works.
P7 – Parameter protection	
Parameters that fix legally relevant characteristics of the measuring instrument shall be secured against unauthorised modification. Required Documentation: The documentation should describe all of the legally relevant parameters, their ranges and nominal values, where they are stored, how they may be viewed, how they are secured and when, i.e. before or after verification.	Checks based on documentation: - Check that changing or adjusting secured device-specific parameters is impossible after securing. - Check whether all relevant parameters according to the lists (given in Extension I, if any) have been classified as secured. Functional checks: - Test the adjusting (configuration) mode and check whether disabling after securing works. - Examine the classification and state of parameters (secured/settable) at the display of the instrument, if a suitable menu item is provided.
L1 – Completeness of measurement data stored	
The measurement data stored must contain all relevant information necessary to reconstruct an earlier measurement. Required Documentation: Description of all fields of the data sets.	Checks based on documentation: - Check whether all information needed for the relevant legal and metrological purposes are contained in the data set.
L2 – Protection against accidental or unintentional changes	
Stored data shall be protected against accidental and unintentional changes. Required Documentation: Description of protection measures (e.g. the checksum algorithm, including the length of the generator polynomial).	Checks based on documentation: - Check that a checksum over data is generated. - Check that legally relevant software, which reads the data and calculates a checksum really compares the calculated and the nominal values. - 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 given to the user when he is about to delete measurement data files. Functional checks: - Check by practical spot checks that a warning is given before deleting measurement data, when deleting is possible at all.
L3 – Integrity of data	
The measurement data stored must be protected against intentional changes. Required Documentation: The method of how the protection is realised shall be documented.	Checks based on documentation: - If a checksum or signature is used o Check that the checksum or signature is generated over the entire data set. o Check that legally relevant software, which reads the data and calculates a checksum or decrypts a signature really compares the calculated and the nominal values. - Check that secret data (e.g. key initial value, if used) are kept secret against spying out by simple tools.

	Functional checks: - Check that a falsified data set is rejected by the retrieval program.
L4 – Authenticity of measurement data stored	
The measurement data stored must be capable of being authentically traced back to the measurement that generated them. Required Documentation: Description of the method used for ensuring the authenticity.	Checks based on documentation: - Check that there is correct linking between each measurement value and the corresponding measurement. - If a checksum or signature is used, check that the checksum or signature is generated over the entire data set. - Check that secret data (e.g. key initial value if used) are kept secret against spying out by simple tools. Functional checks: - Check whether the corresponding stored data and the data printed on the ticket or invoice are identical.
L5 – Confidentiality of keys	
Keys and the accompanying data must be treated as legally relevant data and must be kept secret and be protected against compromise by software tools. Required Documentation: Description of the key management and means for keeping keys and associated information secret.	Checks based on documentation: - Check that the secret information cannot be compromised.
L6 – Retrieval of stored data	
The software used for verifying measurement data sets stored shall display or print the data, check the data for changes, and warn if a change has occurred. Data that are detected as having been corrupted must not be used. Required Documentation: - Description of the functions of the retrieval program. - Description of detection of corruption. - Operating manual for this program.	Checks based on documentation: - Check that retrieval software really compares the calculated and the nominal values. - Check that retrieval software is part of the legally relevant software. Functional checks: - Check whether the program detects corrupted data sets. - Perform spot checks verifying that retrieval provides all necessary information.
L7 – Automatic storing	
The measurement data must be stored automatically when the measurement is concluded. Required Documentation: Confirmation that storing is automatically carried out. Description of the Graphical User Interface.	Functional checks: - Examine by spot checks that the measurement values are stored automatically after measurement or acceptance of measurement is concluded. Check that there are no buttons or menu items to interrupt or disable the automatic storing.
L8 – Storage capacity and continuity	
The long-term storage must have a capacity which is sufficient for the	Checks based on documentation: - Check that the capacity of storage or a formula for

intended purpose. Required Documentation: Description of management of exceptional cases when storing measurement values.	calculating it is given by the manufacturer. - Check that overwriting of data cannot occur before the end of the data storage period foreseen and documented by the manufacturer. Functional checks: - Check that a warning is given to the user when he is about to delete measurement data files (when deleting is possible at all). - Check that a warning is given when the storage is full or removed.
I3-1 – Fault recovery	
The software shall recover from a disturbance to normal processing. Required Documentation: A brief description of the fault recovery mechanism and when it is invoked. Brief description of the related tests carried out by the manufacturer.	Checks based on documentation: - Check whether the realisation of fault recovery is appropriate. Functional checks: - None in addition to those completed during type approval to confirm correct functioning in the presence of defined influence quantities.
I3-2 – Back-up facilities	
There shall be a facility that provides for the periodic back-up of legally relevant data, such as measurement values, and the current status of the process in case of a disturbance. This data shall be stored in non-volatile storage. Required Documentation: A brief description of which data is backed-up and when this occurs.	Checks based on documentation: - Check whether all legally relevant data are saved to non-volatile storage and can be recovered. Functional checks: - None in addition to those completed during type approval to confirm correct functioning in the presence of defined influence quantities.
I3-3 – Indication suitability	
The display of the total energy shall have a sufficient number of digits to ensure that when the meter is operated for 4000 hours at full load (I = Imax, U = Un and PF = 1) the indication does not return to its initial value. Required Documentation: Documentation of the internal representation of the electrical energy register and auxiliary quantities (variable types).	Checks based on documentation: - Check whether storage capacity is sufficient.
I3-4 – Inhibit resetting of cumulative measurement values	
For utility measuring instruments the display of the total quantity supplied or the displays from which the total quantity supplied can be derived, whole or partial reference to which is the basis for payment, shall not be able to be reset during use. Required Documentation:	Checks based on documentation: - Check that cumulative legally relevant measurement values cannot be reset without evidence of intervention. Functional checks: - None in addition to those completed during type approval to confirm correct functioning. Refer to P3 and P4.

Documentation of protection means against resetting the energy registers.	
I3-5 – Dynamic behaviour	
The non-legally relevant software shall not adversely influence the dynamic behaviour of a measuring process. Required Documentation: - Description of the interrupt hierarchy. - Timing diagram of the software tasks. Limits of proportionate runtime for non-legally relevant tasks.	Checks based on documentation: - Check that documentation of the limits of the proportionate runtime for non-legally relevant tasks is available to the programmer of the non-legally software part. Functional checks: - None in addition to those completed during type approval to confirm correct functioning in the presence of defined influence quantities.

Table 10-1: The complete list of requirements and validation guidance applicable to the built-for-purpose meters of active electrical energy

10.2 An overview of common points in the WELMEC Guide 7.2 and the related fields

Quality Characteristics as Defined in ISO/IEC 9126		**P**	**U**	**L**	**S**	**D**	**T**	**I-3**	**Tot.**
Reliability									10
	Maturity	1	1	1			1		4
	Fault Tolerance	1	1			1			3
	Recoverability	1	1				1		3
Usability									1
	Understandability		1						1
	Learnability								
	Operability								
	Attractiveness								
Functionality									61
	Suitability	1	1	5		3	4	2	16
	Accuracy	1	1	5			1		8
	Interoperability	2	2						4
	Security								33
	Security – Authenticity	1	3	1		2	1	1	9
	Security – Integrity	5	5	3		2	2		17
	Security – Confidentiality	2	2	2			1		7
Efficiency									0
	Time Behaviour								
	Resource Utilisation								
Maintainability									4
	Analysability	1	1		1				3
	Changeability				1				1
	Stability								
	Testability								
Portability									2
	Adaptability								
	Installability								
	Co-Existence		1					1	2
	Replaceability								

Table 10-2: Common points in the WELMEC Guide 7.2 and International Standard ISO/IEC 9126

ISO/IEC 12119:1994 **Information technology – Software packages – Quality requirements and testing**	**P**	**U**	**L**	**T**	**S**	**D**	**I-3**	**Tot.**
3.1 Product description 3.1.1 General requirements on contents								
3.1.2 Identifications and indications Identification of the product description								
Identification of the product	P1	U1				D3	I3-6	
Supplier								
Work task							I3-2	
Required system (processor, memory, storage, I/O, network, other SW)	P1	U1	L8					
Interfaces	P1, P3, P4	U1, U3, U4			S3	Dx		
Items to be delivered								
Installation						Dx		
Support								
Maintenance								
3.1.3 Statements on functionality a. Overview of functions	P1	U1	L1		S1	D1, Dx	I3-2, I3-4	
b. Boundary values - min. max. values, - length of keys, - max. no. of records in a file, - max. number of search criteria, - minimum sample size	P7	U7	L1, L8	T1				
c. Security	P2, P5, P6, P7	U1, U2, U3, U4, U5, U6, U7, U8	L2, L3, L4, L5, L6	T2, T3, T4, T5, T6	S3	D1, D2, D3	I3-5	
3.1.4 Statements on reliability			L8	T7, T8		D1	I3-1	
3.1.5 Statements on usability			L1	T1	S2		I3-3, I3-4	
3.1.6 Statements on efficiency								
3.1.7 Statements on maintainability								
3.1.8 Statements on portability								

3.2 User documentation – general	P1	U1							
3.2.1 Completeness									
3.2.2 Correctness									
3.2.3 Consistency									
3.2.4 Understandability									
3.2.5 Ease of overview									
3.3 Programs and data 3.3.1 Functionality –eneral	P3, P4, P5		L2						
Functionality – Installation						Dx			
Functionality – Presence of functions		U2							
Functionality – Correctness			L1, L7	T2, T6	S2, S3	D1	I3-2, I3-3, I3-4		
Functionality – Consistency			L1. L3						
3.3.2 Reliability			L8			D1	I3-1		
3.3.3 Usability Understandability Ease of overview Operability			L1	T1	S2		I3-2, I3-3		
3.3.4 Efficiency									
3.3.5 Maintainability									
3.3.6 Portability									

Table 10-3: Common points in the WELMEC Guide 7.2 and the ISO/IEC 12119:1994

Guidance Document	WELMEC Guide 2.3	WELMEC Guide 2.5	WELMEC Guide 7.2	OIML D-SW	NORDTEST	Measurement Canada	EEEE Guide	FDA SW Validation	FDA 21 CFR Part 11	EN 61508
SW lifecycle addressed					Yes		Yes	Yes		Yes
SW verification					Yes		Yes	Yes		Yes
Requirements specification					Yes			Yes		Yes
Design					Yes			Yes		Yes
Implementation (coding)					Yes			Yes		Yes
Development tips					Yes			Yes		Yes
SW testing techniques				Yes	Yes		Yes	Yes		Yes
System acceptance test					Yes		Yes	Yes		Yes
Suggested validation methods	Yes		Yes	Yes	Yes		Yes	Yes		Yes
SW validation after a change	LM specific	LM specific	LM specific	Yes	Yes	Yes	Yes	Yes		Yes
Indication of results (user interface)			Yes	Yes		Yes				
Virus protection						Yes	Yes			
Examples of acceptable solutions	Yes	Yes	Yes							Yes
Report / certificate on SW examination	Yes	Yes	Yes	Yes	Yes			Yes		
Year of issue	1995	2000	2005	2004	2003	2005	2004	2002	1997	2002/2003
No. of pages	13	21	117	41	13	7	23	47	3	430
Intended use	SW in NAWI, free - programmable PC modules	SW in NAWI, module	SW in LM instruments covered by MID	SW in all LM instruments	Computerised systems in testing and cal. labs.	Metrological. software excluding built for purpose	Computerised systems in testing and cal. labs.	Medical devices	Electronic records and signatures	Safety related systems
Intended users (target audience)	MAN, LMTAA, LMI, LMV	MAN, LMTAA, LMI, LMV	MAN, LMTAA, LMI, LMV	MAN, LMI, LMV	MAN, AB, ULAB	MAN, LMTAA, LMI, LMV	MAN, AB, ULAB	UMDQSR, MAN, UTMDDDP, FDAI, FDAC, AB	UMDQSR, MAN, UTMDDDP, FDAI, FDAC, AB	MAN, TAA
Kind of applications (primarily)	SWWS	SWEMB, SWWS	SWEMB, SWWS	SWEMB, SWWS	SWEMB, SWWS, SWSA, SWAM	SWWS	SWEMB, SWWS, SWSA, SWAM, SWQS	SWEMB, SWWS, SWSA, SWPLC, SWQS, SWAM	SWEMB, SWWS, SWSA, SWPLC, SWQS, SWAM	

Risk evaluation / consideration			**Yes**	**Yes**	**Yes**		**Yes**	**Yes**		**Yes**
Protection against changes of SW and data	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**			
Protection of SW	**Yes**		**Yes**	**Yes**	**Yes**	**Yes**				**Yes**
SW update (download)	**Yes**		**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**		
SW separation	**Yes**		**Yes**	**Yes**						**Yes**
Transmission		**Yes**	**Yes**	**Yes**		**Yes**	**Yes**		**Yes**	**Yes**
Storage		**Yes**	**Yes**	**Yes**			**Yes**	**yes**	**Yes**	**Yes**
SW and / or data integrity			**Yes**	**Yes**		**Yes**	**Yes**		**Yes**	
Confidentiality			**Yes**	**Yes**			**Yes**		**Yes**	
Authenticity			**Yes**	**Yes**			**Yes**		**Yes**	
Distributed meas. systems (network)		**Yes**	**Yes**	**Yes**		**Yes**	**Yes**		**Yes**	
SW / record identification	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Interfaces	**Yes**	**Yes**	**Yes**	**Yes**				**Yes**		**Yes**
Storage capacity		**Yes**	**Yes**	**Yes**						
Event logger (audit trail)	**Yes**		**Yes**	**Yes**	**Yes**	**Yes**	**Yes**		**Yes**	

Table 10-4: An overview of issues covered by the analysed documents

Categories of Users		Object of Validation	
MAN	Persons responsible for the design, development, or production of medical device software (Manufactures of LM measuring instrument)	SWEMB	Software used as a component, part, or accessory of a medical (measuring) device – embedded
UMDQSR	Persons subject to the medical device Quality System regulation	SWWS	Software used as a component, part, or accessory of a medical (measuring) device – workstation based
UTMDDDP	Persons responsible for the design, development, production, or procurement of automated tools used for the design, development, or manufacture of medical devices or software tools used to implement the quality system itself	SWQS	Software used in implementation of the device manufacturer's quality system (e.g. software that records and maintains the device history record)
FDAI	FDA Investigators	SWPLC	Software used in the production of a device (e.g. programmable logic controllers in manufacturing equipment)
LMI	Legal Metrology Inspection (Metrological Surveillance)		
LMV	LM Verification		
FDAC	FDA Compliance Officers	SWAM	Software for Automation of Measurements
LMTAA	Type Approval Authority	SWSA	Software that is itself a medical (measuring) device
AB	Accreditation Body		
ULAB	Laboratory Users		

Table 10-5 : Explanation of abbreviations in Table 10-4

CA Module	Conformity Assessment Activity		Software Testing and Related Activities	
	Manufacturer	**Conformity Assessment Body**	**Manufacturer**	**Conformity Assessment Body**
B	**B.1.1.** **Verification of the development phase (type testing)**	**B.1.2** **Type examination (approval)** **WELMEC Guide 8.3**	**B.1.3** **Refinement of the requirements.** **Preparation of the test plan.** **Software lifecycle activities.** **Documentation.** **Unit testing, function testing, system testing, acceptance testing.** **Volume testing, stress testing, usability testing, security testing, performance testing, storage testing, configuration testing, compatibility testing, installability testing, reliability testing, recovery testing, serviceability testing, documentation testing, incremental and regression testing.** **Black-box testing strategies: e.g.: equivalence partitioning, boundary value analysis.** **White-box testing strategies: inspection, walkthrough.** **Applied standards and knowledge base:** **ISO/IEC 9126-1:2001, Software Engineering – Product Quality – Part 1: Quality Model,** **ISO/IEC 12119:1994, Software Packages – Quality Requirements and Testing,** **Myers, J. G.: The Art of Software Testing; John Wiley & Sons, Inc., 1979,** **ISO/IEC 6592:1999, Information Technology – Guidelines for the Documentation of Computer -Based Application Systems,** **ISO/IEC 9127:2001, Software Engineering – User Documentation and Cover Information for Consumer Software Packages,** **ISO/IEC 15408: 1999: Common Criteria for Information Technology Security Evaluation (CC, Version 2.1),** **ISO/IEC 12207: 1995, Information Technology – Software Life Cycle Processes,** **ISO/IEC 14598 – Software Engineering – Product Evaluation,** **BS 7925 -Software Component Testing,** **EN 61508, Functional safety of electrical / electronic / programmable electronic safety-related systems, Parts 1–7, 2002–2003.**	**B.1.4** **Application of the WELMEC Guide 7.2**
D	**D.1.1** **Approval of quality system**	**D.1.2.** **Audit of the production and**	**D.1.3** **Final testing (functional, black-box test).**	**D.1.4** **/**

	for production	**final testing** **ISO 9001:2000** **WELMEC Guide 8.4**		
F	**F.1.1** **/**	**F.1.2** **Product verification**	**F.1.3** **/**	**F.1.4** **Functional (black-box) test**
G	**G.1.1** **/**	**G.1.2** **/**	**G.1.3**	**G.1.4** **Thorough functional and design examination[1], may consist of several instances of B.1.3**
H1	**H1.1.1** **Development, construction** **Type testing** **Type examination (approval[2])**	**H1.1.2** **Design Examination** **WELMEC Guide 8.2**	**H1.1.3** **Besides the activities of B.1.3 in the scope of type testing:** **Implementation of the software configuration management:** **ISO/IEC 15504: 2004: Software Engineering -Process Assessment (SPICE),** **U.S. Department Of Health and Human Services, Food and Drug Administration: General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002.**	**H1.1.4** **Audit of the production final testing, development and type testing;** **WELMEC Guide 8.2**

Table 10-6: Linkage between conformity assessment modules and software validation activities

[1] Applicable to non mass-produced single instruments.

[2] The term "type approval" was used before in legal metrology, MID deals with the term "type examination".

MID Clause	Generic Property	Concrete Functionality / Requirement	WELMEC Guide 7.2
4(b)	Definitions, Arrangement of sub-assemblies	Transmission of legally relevant data. Basic Guides applicable to sub-assemblies.	T P, U
10	Technical documentation	Documentation of design, manufacture and operation. Enable assessment of conformity. General description of the instrument. Description of electronic devices with drawings, flow diagrams of the logic, general software information. Location of seals and markings. Conditions for compatibility with interfaces and sub-assemblies.	P1, U1
AI-6	Reliability	Fault detection, back-up, restoring, restart.	I1-1 to I1-3, I2-1 to I2-3, I3-1 to I3-3, I4-1 to I4-3, I6-1 to I6-2
AI-7	Suitability	No features to facilitate fraudulent use. Minimal possibilities for unintentional misuse.	P3 - P7, U3 - U8, L1 – L5, L7, L8 T1 – T8, S2, D3, D4
AI-8	Protection against corruption		
AI-8.1		No influences by the connection of other devices.	P4, U4
AI-8.2		Securing; evidence of intervention.	P6, P7, U6, U7, D1, D4
AI-8.3		Identification of software; evidence of intervention.	P2, P6, P7, U2, U6, U7, U8, D2, D4
AI-8.4		Protection of stored or transmitted data.	P5 - P7, U5 - U7, L1 - L5, T1 - T8 D1 - D3
AI-8.5		No reset of cumulative registers.	I1-5, I2-5, I3-5, I4-5
AI-9.1		Measuring capacity (rest of items not relevant to software).	L8
AI-9.3		Instructions for installation, ..., conditions for compatibility with interface, subassemblies or measuring instruments.	P1, U1
AI-10.1		Indication by means of a display or a hard copy.	U8, L6, S2
AI-10.2		Significance of result, no confusion with additional indications.	U8, L1, L4, L6, S2
AI-10.3		Print or record easily legible and nonerasable.	U8, L6, S2
AI-10.4		For direct sales: presentation of the result to both parties.	U8, S2
AI-10.5		For utility meters: display for the customer.	I1-6, I2-6, I3-6, I4-6

MI-001-7.1.1, MI-001-7.1.2	**Electromagnetic immunity**	**Fault detection. Back-up facilities. Wake-up facilities and restoring.**	**I1-1 to I1-3**
MI-002-3.1	**Electromagnetic immunity**	**Fault detection. Back-up facilities. Wake-up facilities and restoring.**	**I2-1 to I2-3**
MI-002-5.2	**Suitability**	**Acceptable solution for monitoring battery lifetim.e**	**I2-7**
MI-002-5.3	**Suitability**	**Internal resolution**	**I2-4**
MI-002-5.5	**Suitability**	**Test element.**	**I2-9**
MI-002-9.1	**Volume conversion devices Suitability**	**Acceptable solution for monitoring the gas volume converter.**	**I2-8**
MI-003-4.3	**Permissible effect of transient electromagnetic phenomena**	**Fault detection. Back-up facilities. Wake-up facilities and restoring.**	**I3-1 to I3-3**
MI-003-5.2	**Suitability**	**Internal resolution.**	**I3-4**
MI-004-4.2	**Permissible influences of electromagnetic disturbances**	**Fault detection. Back-up facilities Wake-up facilities and restoring.**	**I4-1 to I4-3**
MI-006-IV, MI-006-V	**Discontinuous and continuous Totalisers**	**Fault detection Back-up facilities.**	**I6-1 to I6-2**

Table 10-7: An overview of the refinement of MID requirements in WELMEC Guide 7.2

Methods Suggested by WELMEC Guide 7.2	**P**	**U**	**L**	**T**	**S**	**D**	**I-3**	**Tot.**
Analysis of the documentation	P2, P3, P4, P5, P6, P7	U2, U3, U4, U5, U6, U7, U8	L1, L2, L3, L4, L5, L6, L8	T1, T2, T3, T4, T5, T6, T7, T8	S1, S2, S3	D1, D2, D3, D4	I3-1, I3-2, I3-3, I3-4, I3-5	40
Functional check (regular use)	P2, P5, P7	U2, U5, U7; U8	L2, L4, L6, L7, L8	T8	S2	D1, D4	I3-1, I3-2, I3-4, I3-5	28
Functional check (regular and irregular parameters)	P3, P4, P6	U3, U4	L3, L6	T6		D2, D3		10
Data flow analysis [54]	P3							1
Source code analysis[3]	P4, P5, P6, P7	U2, U3, U6, U7, U8	L1, L2, L3, L4, L5, L6, L7, L8	T1, T2, T3, T4, T5, T6, T7, T8	S1, S2, S3	D1, D2, D3, D4		33

Table 10-8: Software validation methods applicable to particular requirements of WELMEC Guide7.2

Pi = Basic configuration P requirements,
Ui = Basic configuration U requirements,
Li = Extension L requirements,
Ti = Extension T requirements,
Si = Extension S requirements,
Di = Extension D requirements,
Ii = Extension I requirements.

[3] For higher risk classes.

Methods suggested by MID-SW WP2		
Methods based on checking the documentation	Methods based on functional checks	Methods based on checking the source code
Identification of the software	Functional testing	Software design review (inspection)
Completeness of the documentation	Simulation of input signals	Review of software documentation
Examination of the operating manual and technical documentation		Data flow analysis
Inspection of the specification		White-box testing
Field experience		State transition diagram
		Protocol analysis
		Testing of software modules
		FMEA (Failure Mode and Effect Analysis)
		Event tree analysis
		Fault tree analysis
		Analysis of fault detection (program sequence monitoring: "watchdog")

Table 10-9: An overview of methods suggested by MID-SW WP2

10.3 List of participants in the MID-Software Project

Physikalisch – Technische Bundesanstalt (PTB)	Germany	National metrology institute
Delta (Danish Electronics, Light & Acoustics)	Denmark	Notified body
Gum (Główny Urząd Miar)	Poland	National metrology institute
HALE Electronic	Austria	Manufacturer of measuring instruments (taximeters)
The Herbert Group	United Kingdom	Manufacturer of measuring instruments (weighing instruments)
Instituto Português Da Qualidade	Portugal	National metrology institute
Justervesenet (JV)	Norway	National metrology institute
Gilbarco Veeder-Root	Italy	Manufacturer of measuring instruments (Fuel dispensers)
Mettler Toledo	Switzerland	Manufacturer of measuring instruments (weighing instruments)
Nederlands Meetinstituut (NMi)	The Netherlands	National metrology institute
National Weights And Measures Laboratory (NWML)	United Kingdom	Notified body
Sartorius Mechatronics	Germany	Manufacturer of measuring instruments (weighing instruments)
Landis & Gyr	Switzerland	Manufacturer of measuring instruments (electric energy meters)
Metrology Institute of the Republic of Slovenia (MIRS)	Slovenia	National metrology institute
Swedish National Testing and Research Institute (SP)	Sweden	National metrology institute
Laboratoire national de métrologie et d'essais (LNE)	France	National metrology institute

10.4 Common and most frequent errors

Inspection Error Checklist Summary							
Data Reference	Computation	Interfaces	Data Declaration	Comparison	Control Flow	Input/Output	Other Cheeks
1. Unset variable used?	1. Computations on nonarithmetic variables?	1. Number of input parameters equal to number of arguments?	1. All variables declared?	1. Comparisons between inconsistent variables?	1. Multiway branches exceeded?	1. File attributes correct?	1. Any unreferenced variables in cross-reference listing?
2. Subscripts within bounds?	2. Mixed-mode computations?	2. Parameter and argument attributes match?	2. Default attributes understood?	2. Mixed-mode comparisons?	2. Will each loop terminate?	2. OPEN statements correct?	2. Attribute list what was expected?
3. Non integer subscripts?	3. Computations on variables of different lengths?	3. Parameter and argument units system match?	3. Arrays and strings initialized properly?	3. Comparison relationships correct?	3. Will program terminate?	3. Format specification matches I/O statement?	3. Any warning or informational messages?
4. Dangling references?	4. Target size less than size of assigned value?	4. Number of arguments transmitted to called modules equal to number of parameters?	4. Correct lengths, types, and storage class as assigned?	4. Boolean expressions correct?	4. Any loop bypasses because of entry conditions?	4. Buffer size matches record size?	4. Input checked for validity?
5. Correct attributes when aliasing?	5. Intermediate result overflow or underflow?	5. Attributes of arguments transmitted to called modules equal to attributes of parameters?	5. Initialization consistent with storage class?	5. Comparison and Boolean expressions mixed?	5. Are possible loop fall-throughs correct?	5. Files opened before use?	5. Missing function?
6. Record and structure attributes match?	6. Division by zero?	6. Units system of arguments transmitted to called modules equal to units system of parameters?	6. Any variables with similar names?	6. Comparisons of base-2 fractional values?	6. Off-by-one iteration errors?	6. Files closed after use?	
7. Computing addresses of bit strings? Passing bit-string arguments?	7. Base-2 inaccuracies?	7. Number, attributes, and order of arguments to built-in functions correct?		7. Operator precedence understood?	7. DO/END statements match?	7. End-of-file conditions handled?	
8. Based storage attributes correct?	8. Variable's value outside of meaningful range?	8. Any references to parameters not associated with current point of entry?			8. Any nonexhaustive decisions?	8. I/O errors handled?	
9. Structure definitions match across procedures?	9. Operator precedence understood?	9. Input-only arguments altered?			9. Any textual or grammatical errors in output information?		
10. Off-by-one errors in indexing or subscripting operations?	10. Integer divisions correct?	10. Global variable definitions consistent across modules?					
11. Are inheritance requirements met?		11. Constants passed as arguments?					

Table 10-10: Checklist of common and most frequent errors

10.5 Most important issues for system testing

System Testing: Most Important Elements	
Facility Testing	Test that the program has actually implemented all the required functions.
Volume Testing	Test in which the program is loaded with an extremely large amount of data (for example, the linker receives the task to make an executive file of 10,000 software modules).
Stress Testing	Test in which the program is loaded with a large amount of input data in the shortest possible time (applicable to real-time systems – e.g., when an air traffic control system with a capacity of 200 aircraft simulates the simultaneous entrance of 200 or more aircraft in its airspace).
Usability Testing	Test in which the user interface (customer service access point) is tested against: • serving possibility (especially for real-time systems), • sufficiency of the output data, the required intelligence, education or skills of the user for normal use of the program, • reasonability (understandability) of the messages and results of the program, • existence of unused functions.
Security Testing	Testing the protection of program and data from unauthorised users (especially for network applications).
Performance Testing	Testing the response time and throughput in the conditions defined by the specification.
Storage Testing	Testing of the use of memory resources.
Configuration Testing	Testing the operation of the program in all required configurations (different number of networked computers, communication connections, operating systems, etc.).
Compatibility/Conversion Testing	Testing the compatibility with earlier versions of the program (e.g., that the database correctly interprets the data generated by the previous version).
Installability Testing	Testing the installation on various configurations of hardware.
Reliability Testing	All failures during the testing are recorded due to the introduction in the reliability model, which enables the prediction of program behaviour during use.
Recovery Testing	Testing the compliance with the requirements for the restoration of the system.
Serviceability Testing	Testing the suitability for maintenance (diagnostics, maintenance procedures, documentation).
Documentation Testing	Testing the comprehensiveness of user documentation.
Procedure Testing	Testing the procedures foreseen to be performed by the user of the program, or which should be carried out by the user or third parties (such as a database administrator).

Table 10-11: System testing: most important elements

10.6 Elements of a software test plan

Objectives	**Objectives of every phase have to be clearly determined**
Test completion criteria	The criteria for completion of every phase should be clearly defined.
Timetable/schedule	It is necessary to define the timetable of activities (planning, design and implementation of test cases).
Responsibilities	For each phase of testing it is necessary to define the human resources for planning, design and implementation of test cases, and debugging.
Test case libraries, standard test cases	It is very useful – especially in large projects – if the test cases are systematically recorded and stored (for regression testing, possible use in other related projects, ...).
Tools	It is necessary to select or design the necessary tools,and plan their use in time.
Processor time	This is an ancient requirement originating from the times when the processor time on a mainframe computer had to be shared between several users. Today, given the fall of the prices of hardware, it is less important, except for very demanding applications.
Configuration of the hardware	During the testing it is necessary to ensure the configuration of hardware and software (environment), which will match the foreseen configuration of the program in use. Purchases of individual items have to be time-optimized.
Integration	The test plan must be coordinated with the integration plan. It is necessary to provide facilities for preparation of the substitution modules.
Tracking procedures	Monitoring the various aspects of testing: identification of the critical modules, estimation of the progress as regards the time schedule, resources and criteria for completion.
Debugging procedures	Defining the procedures for reporting the errors, tracking the corrections and adding the corrections into the system. In addition, it is necessary to plan schedules, responsibilities, tools and resources.
Regression testing	Practice has shown that that, during the removal of an error, some new error will be entered (according to some authors from 0.1 to 0.6 new errors per removal of one error). Therefore it is necessary to check the previous functionality of the program after every set of corrections.

Table 10-12: Elements of the test plan

10.7 An example table for the classification of the legal relevance of parameters for electricity meters[4]

Functionality	Examples	Covered by MID	Settable (not protected by verification seal)	Protected (under verification seal)	Remark
Basic characteristics of the instrument	Accuracy class, nominal voltage and current, frequency, I/O configuration	x		x	Usually also strongly influenced by hardware.
Energy registers (basic measurement)	Total and rated energy registers including their representation and identification on the display, meter constant	x		x	
Error register		x		x	
Security system	Access rights to data and parameters	x		x	When available.
Calendar clock	Time/date, DST settings, clock base		x [1)]		Not used for energy measurement.
Demand registers	Average demand, maximum demand, cumulative maximum demand, integration period		x [1)]		
Stored values profile or other profiles	Capture period, buffer size, captured registers	x [2)]	x		
Rate control facilities	TOU and other control functions used for rate control		x		
Communication settings	Transmission speed, protocol settings, passwords		x		

Table 10-13: Classification of the legal relevance of parameters

[1)] These functions are not covered by MID. Anyway, it is proposed to protect them under the verification seal, if they are used for billing.

[2)] Extension L is only applicable if the profiles are used for billing. Therefore the same rule as under[1)] is proposed.

All other functions of the instrument are not covered by MID and can therefore be changed after verification. The protection usually consists of a utility seal or a password.

[4] Proposed by the representative of Landis+Gyr during the meeting of the Group for Analysis of the Results of the Comparative Software Valdiation Experiment.

10.8 Most important terminology

Most important terminology used in the thesis	
Conformity assessment (CA)	Activities aimed to determine, directly or indirectly, that a process, product, or service meets relevant standards and fulfils relevant requirements.
Conformity assessment module	Specific set of agreed conformity assessment activities related to: - a particular phase of product development or production, - whether the final product is suitable for serial production, - the implemented quality assurance system and organisational structure of the manufacturer, - the selected CA activites (e.g. type testing) and performers of CA activities (e.g. in own or third-party laboratory).
Validation	Confirmation by examination and provision of objective evidence that the particular requirements for a specific intended use are fulfilled.
Software validation	Software validation is confirmation by examination and provision of objective evidence that software specifications conform to user needs and intended uses, and that the particular requirements implemented through software can be consistently fulfilled.
Testing	Determination of the characteristics of a product, a process or a service according to certain procedure, methodology or requirements. Testing is the most important element of validation.
Software testing	The process of exercising the software to prove that it satisfies specified requirements and to detect errors, i.e.: The process of operating a system or component under specified conditions, observing or recording the results, and making an evaluation of some aspect of the system or component. The process of analysing a software item to detect the differences between existing and required conditions and to evaluate the features of the software items.
Software Fault	An accidental condition that causes a functional unit to fail to perform its required function. A defect that causes a reproducible or catastrophic malfunction. A malfunction is considered reproducible if it occurs consistently under the same circumstances.
Software Failure	State of the system when the delivered service no longer complies with the specifications, the latter being an agreed description of the system's expected function and/or service.
Software Error	Part of the system which is liable to lead to subsequent failure.
Verification	Confirmation by examination and provision of objective evidence that specified particular requirements (e.g. for certain development phase) have been fulfilled.
Type testing (Type examination)	Testing of relevant quality characteristics (regarding essential requirements of binding legislation and technical standards, elements of technical specification and other declared functionality of a product) that are declared to be specific for one type of product. Type testing is perfomed on samples that are declared to be representative

	of a type.
Type approval	Permission for a produst or process to be marketed or used for stated purposes or under stated conditions.
Verification (legal metrology)	Procedure (other than type approval) which includes the examination and marking and/or issuing of a verification certificate, that ascertains and confirms that the measuring instrument complies with the statutory requirements (or approved type of instrument). Legal verification is performed on each unit from the population of some type of instrument (as opposed to type testing, which is performed on the representatives of a type of instrument). Verification may be: - initial (before the first use of an instrument) - regular (checking the measuring instrument's conformance with the statutory requirement in regular time intervals) - extraordinary (e.g. after a maintenance intervention that affects a significant metrological functionality of a measuring instrument)
Surveillance	Systematic iteration of conformity assessment activities as a basis for maintaining the validity of the statement of conformity.
Inspection	Examination of a product design, product, process or installation and determination of its conformity with specific requirements or, on the basis of professional judgment, with general requirements.
Auditing	Systematic, independent, documented process for obtaining records, statements of fact or other relevant information and assessing them objectively to determine the extent to which the specified requirements are fulfilled.
Review	Verification of the suitability, adequacy and effectiveness of selection and determination activities, and the results of these activities, with regard to fulfilment of specified requirements by an object of conformity assessment.
Attestation	Issue of a statement. Based on a decision following review, that fulfilment of specified requirements has been demonstrated.
Certification	Third-party attestation related to products, processes, systems or persons.
Accreditation	Third-party attestation related to a conformity assessment body conveying formal demonstration if its competence to carry out specific conformity assessment tasks.
Notified body	A notified body is an independent body appointed by a notifying body (from an EU Member State) as being capable of performing conformity assessment according to a specific Directive.
Notifying body	Institution designated by the State for notifying the entities capable of conformity assessment according to a specific Directive to the EU Commission (e.g. in Slovenia, the Ministry of Economy is the notifying body for the Directive on Measuring Instruments is).

POVZETEK

Predstavljena disertacija analizira, razvija in vrednoti nove znanstvene pristope k validaciji programske opreme merilnih instrumentov, ki se uporabljajo kot zakonska merila. Vzporedno z znanstvenimi in tehničnimi vidiki so predstavljeni vidiki in specifike zakonskega meroslovja. Posebna pozornost in skrb sta v zakonskem meroslovju bili posvečeni programski opremi zaradi številnih težav (ugotovljenih poskusov goljufij, ki so bile realizirane v programski opremi merilnih instrumentov, ki se uporabljajo v prometu blaga in storitev[1]).

Za uporabo pravilnih metod validacije programske opreme v zakonskem meroslovju je nujno poznavanje njenih svojstvenih lastnosti. Te so vezane na:

- specifiko merilne tehnike (podpora visoki točnosti, zanesljivosti, razpoložljivosti, varnosti, obnovljivosti in ponovljivosti meritev),
- tehnološko realizacijo (zgradba merilnih sistemov iz komponent različnih dobaviteljev z uporabljenimi različnimi tehnologijami),
- zmanjševanje stroškov uporabe in vzdrževanja,
- podporo zaščiti potrošnikov (preprečevanje zlorab),
- podporo novim udeležencem v procesih zakonskega meroslovja,
- podporo oddaljenim overitvam in inšpekcijskim pregledom,
- kompatibilnost z lastno zgrajenimi komponentami,
- nenazadnje, na podporo dodatni uporabniški funkcionalnosti, ki omogoča konkurenčno prednost.

Teoretični in praktični rezultati so bili implementirani v vodilu WELMEC 7,2 "Software Guide (Measuring Instrument Directive 2004/22/EC)"[41]. Poleg tega je bila razvita in validirana metoda za potrditev uporabnosti vodila, in sicer z metodo primerjalne validacije enakega vzorca programske opreme med več laboratoriji. Delo lahko šteje kot pred-normativna raziskava na področju standardizacije meroslovne programske opreme. Cilj disertacije je zagotavljanje splošnih rešitev za ugotavljanje skladnosti pravil in postopkov, ne samo v zakonskem meroslovju, temveč tudi za druge tehnične sisteme, kjer pomembno vlogo igra programska oprema.

Disertacija vsebuje osem poglavij.

[1] Mednarodna organizacija zakonskega meroslovja (Organisation Internationale de Métrologie Légale - OIML) je že leta 1999 organizirala seminar na temo programske opreme v merilnih instrumentih (Seminar on software in measuring instruments, 30.09. – 01.10.1999).

1 Opis področja, na katerega se nanašajo originalni prispevki k znanosti

Zakonsko meroslovje je celota tehničnih in upravnih postopkov, določenih z zakonodajo, z namenom zagotavljanja kakovosti meritev na področjih:

- varovanja zdravja ljudi in živali,
- varstva okolja in splošne tehnične varnosti,
- prometa blaga in storitev,
- postopkov pred upravnimi in pravosodnimi organi.

Zakonska merila so v svojem življenjskem ciklu podvržena številnim postopkom ugotavljanja skladnosti. Predpogoj za pridobivanje statusa zakonskega merila je uspešno zaključen postopek odobritve tipa. Pred prvo uporabo merila, ki ima odobritev tipa, je obvezna t.i. prva overitev merila. Za merila v uporabi so obvezna redna preverjanja skladnosti (t.i. redne overitve) in izredne overitve (npr. po popravilu merila). Neodvisno od teh postopkov se izvajajo tudi inšpekcijski pregledi meril v uporabi. Postopki ugotavljanja skladnosti so porazdeljeni med državne institucije (urade, inšpekcijske službe), javne zavode in zasebne, gospodarske družbe. Kakovost njihovega dela nadzorujejo nacionalne akreditacijske službe.

Meroslovje, kot znanstvena disciplina, je povezano z vsako aktivnostjo človeškega življenja. Meritve spremljajo vsako človeško aktivnost, začenši z merjenjem časa za nastavitev budilke, merjenjem porabe vode v kopalnici in kuhinji, porabljene energije za pripravo jutranje kave, hitrosti avtomobila na poti do službe, količine bencina na bencinski postaji – številne meritve spremljajo vsak korak vsakega dne človeškega življenja. Zaradi svojega pomena morajo biti meritve ustrezno koordinirane na vseh nivojih, od meritev, ki omogočajo vsakdanje življenje, do mednarodnih etalonov najvišje točnosti. Zato je meroslovje na mednarodni ravni zelo dobro organizirano. Mednarodne (BIPM [50], OIML [51]) in regionalne meroslovne institucije (EURAMET[72], EURACHEM[73], EUROLAB[74], EA[75], WELMEC [52], COOMET[76]) skrbijo za ustrezno meroslovno infrastrukturo, postopke in navodila za zagotavljanje sledljivosti meritev, postopke ugotavljanja skladnosti in številne druge meroslovne aktivnosti.

1.1 Ugotavljanje skladnosti merilnih instrumentov

Sodobni pristop k ugotavljanju skladnosti daje poudarek na prostem pretoku blaga, element katerega je tudi t.i. "one-stop testiranje", kar proizvajalcem zmanjšuje stroške in skrajšuje čas do dajanja proizvoda na trg.

Istočasno je izjemno pomembno ohraniti primerno raven varstva potrošnikov, kot tudi zagotoviti pogoje za pošteno poslovanje novim podjetjem, katerim se odpirajo nove poslovne priložnosti, ki jih omogoča razvoj merilne tehnike. Primer nove poslovne dejavnosti so posredniki merilnih rezultatov, ki dobaviteljem električne energije posredujejo podatke o porabi, ki jih pridobijo z odčitavanjem porabe pri odjemalcih električne energije.

Predpogoj za doseganje urejenih razmer in zaščito vseh udeleženih strani je zagotavljanje enakovrednosti izvedenih postopkov ugotavljanja skladnosti med vsemi institucijami, ki jih izvajajo. Tak pristop ima dodaten gospodarski vpliv, ker ščiti odgovorne proizvajalce merilnih instrumentov (tiste, ki se ne poslužujejo nekorektnih prijemov za doseganje večjega tržnega deleža).

Dejavnosti, katerih cilj je neposredno ali posredno ugotavljanje, da postopek, izdelek ali storitev izpolnjuje zahteve določenih standardov in druge ustrezne zahteve, se s skupnim imenom imenujejo "ugotavljanje skladnosti". Izvajajo jih organi za ugotavljanje skladnosti. Dejavnosti ugotavljanja skladnosti lahko zajemajo [10]:

- preskušanje,
- nadzor,
- inšpekcijo,
- revidiranje,
- certificiranje,
- registracijo,
- akreditacijo.

V skladu z veljavno evropsko zakonodajo je pogoj za dajanje proizvoda na trg izpolnjevanje bistvenih zahtev vseh ustreznih tehničnih direktiv in z njimi povezanih tehniških standardov na reguliranem področju. V primeru merilnih instrumentov gre za zakonodajo, ki skrbi za:

- varnost uporabnikov merilnih instrumentov,
- varstvo okolja in
- kakovost meritev na področju zakonskega meroslovja.

Varnost uporabnikov zahteva LVD direktiva [68], za varstvo okolja skrbita EMC [69] in RoHS [70] direktivi, meroslovne vidike pa pokriva MID [4] direktiv ali sorodni nacionalni predpisi.

Izpolnjevanje zahtev določenih z zakonodajo je predpogoj za dajanje izdelka na trg in je za uporabnika samoumevno. Zato se uporabniki tega vidika kakovosti praviloma ne zavedajo. Tudi izpolnjevanje zahtev zapisanih v tehnični se jemlje kot samoumevno.

Proizvajalci merilnih instrumentov se zavedajo, da so dodatne funkcije, ki presegajo osnovne zakonske zahteve (ki se nanašajo na varnost, varstvo okolja, pravične trgovine in varovanja zdravja) tiste, ki najbolj povečujejo konkurenčnost njihovih izdelkov.

Kakovost programske opreme torej temelji na funkcionalnosti, ki je namenjena izpolnjevanju osnovnih zahtev, določenih z "zakonodajo", in dodatne funkcionalnosti za doseganje konkurenčne prednosti. Povzetek je prikazan v tabeli 1.

Zahteve kakovosti	**Direkten pomen za uporabnike (in zavedanje)**	**Ugotavljanje skladnosti izvaja**	**Nadzor**
Dodatna funkcionalnost, uporabnost, prijaznost uporabe, vzdrževalnost	Velik	Laboratorij proizvajalca, laboratorij tretje stranke	Tržne zakonitosti
Izpolnjevanje tehnične specifikacije	Šteje kot samoumevno	Laboratorij proizvajalca, laboratorij tretje stranke	Tržne zakonitosti
Bistvene zahteve, ki izhajajo iz zakonodaje (varnost, zaščita okolja, bistvene meroslovne lastnosti)	Ga skoraj ni, razen v primeru odpovedi z resnimi posledicami	Laboratorij proizvajalca, laboratorij tretje stranke, priglašeni organ	Državne institucije (inšpekcijski organi)

Tabela 1: Zahteve kakovosti merilnih instrumentov

Skladno s tako imenovanim »novim pristopom« k ugotavljanju skladnosti obstajajo različni načini, »moduli«, v sklopu katerih pristojne institucije pregledujejo končni izdelek ali proces razvoja in proizvodnje tega izdelka. V primeru števcev električne energije, kar je obravnavani konkretni primer v tej disertaciji, so predvideni načini ugotavljanja skladnosti z zahtevami direktive o merilnih instrumentih:

- B + F: Priglašeni organ izvede pregled tipa instrumenta (na reprezentativnih vzorcih za določen tip). Za vsak posamezni instrument iz serijske proizvodnje se pred prvo uporabo izvede postopek prve overitve,

ali

- B + D: Priglašeni organ izvede pregled tipa instrumenta (na reprezentativnih vzorcih za določen tip). Za vsak posamezni instrument iz serijske proizvodnje proizvajalec deklarira skladnost z odobrenim tipom na podlagi notranje kakovost proizvodnega procesa,

ali

- H1: Priglašeni organ pregleda in odobri sistem zagotavljanja kakovosti (vključno z razvojem in proizvodnjo) proizvajalca. Dodatno priglašeni organ pregleda in odobri konstrukcijo vsakega posameznega tipa instrumenta.

2 Vpliv razvoja informacijske tehnologija na meroslovje

Drugo poglavje podaja pregled vpliva razvoja informacijske tehnologija na meroslovje. Poznavanje stanja tehnike meroslovnih programskih aplikacij je izjemno pomembno za pravilen pristop k njeni validaciji, na primer za oblikovanje zahtev primernih za preskušanje, določanje razreda tveganja in izbire preskusnih metod. Dodatno, poznavanje zgradbe aplikacij lahko olajša druge postopke ugotavljanja skladnosti, kot so overjanje in meroslovni nadzor.

Programska oprema je, tako kot povsod v sodobni tehniki, postala nepogrešljiva tudi v meroslovju. Delovanje sodobnih merilnih instrumentov sloni na vgrajeni programski opremi, nad pridobljenimi merilnimi rezultati se izvajajo številne obdelave. Poleg programov, ki se izvajajo na merilnih instrumentih, uporabljajo sodobni merilni sistemi številne komunikacijske storitve in podatkovne baze. Poleg zbiranja, shranjevanja in obdelave merilnih rezultatov omogoča programska oprema tudi vzdrževanje, kalibracije in inšpekcijski nadzor merilnih instrumentov. Posebnosti programske opreme v zakonskem meroslovju izhajajo iz sledečih zahtev in dejstev:

- podpora visoki točnosti, zanesljivosti, obnovljivosti in ponovljivosti meritev,
- podpora zanesljivosti, varnosti in razpoložljivosti merilnih rezultatov,
- raznolikost hkrati uporabljenih inženirskih tehnologij,
- omogočanje zgradbe merilnih sistemov iz komponent različnih dobaviteljev (ki niso vedno funkcionalno in komunikacijsko združljivi),
- zmanjševanje stroškov dajanja v uporabo in vzdrževanja zakonskih meril (npr. z omogočanjem oddaljene nadgradnje programske opreme meril v uporabi),
- podpora ustrezni zaščiti potrošnikov (zaradi morebitnih zlorab pri merilih, ki se uporabljajo za obračun v prometu blaga in storitev),
- podpora novim udeležencem v procesih zakonskega meroslovja (t.i. ponudnikov merilnih rezultatov),
- omogočanje uporabe in podpora validaciji komponent programske opreme, ki so jo razvili strokovnjaki s področja meroslovja (ki niso strokovnjaki za programsko opremo),
- podpora postopkom overjanja,
- podpora postopkom inšpekcijskih pregledov.

Uporaba sodobnih rešitev informacijske tehnologije na področju merilne tehnike je nedvomno koristna za uporabnike, saj omogoča hitrejše meritve, manjšo merilno negotovost in odpira možnosti za različne analize in nadaljnjo obdelavo. Za proizvajalca nova tehnologija poenostavlja uresničevanje kompleksnih funkcij in vzdrževanje ter daje fleksibilnost za izpolnjevanje dodatnih uporabniških zahtev. Dostop do merilnih instrumentov v obratovanju imajo številni uporabniki: končni odjemalci, dobavitelji (npr. elektrike), lastniki infrastrukturne povezave, vzdrževalci meril, inšpekcijske službe, proizvajalec… Ob upoštevanju števila vključenih strani je jasno, da je tak sistem treba ustrezno koordinirati, spremljati in nadzorovati.

Uporabnost oddaljenega dostopa do merilnih instrumentov ponazorimo na primeru števcev električne energije. Za distributerje električne energije se z oddaljenim dostopom bistveno zmanjšajo stroški zbiranja podatkov o porabi in vzdrževanja merilnih instrumentov. Največje znižanje stroškov vzdrževanja se dosega z možnostjo daljinske nadgradnje programske opreme števcev, ki so že nameščeni na mestu uporabe. V primeru, da se odkrije resna napaka v programski opremi merila, je potrebno programsko opremo veh števcev posodobiti. Za števce električne energije, nameščene v oddaljenih gorskih območij, bi bili lahko stroški obiska servisnega tehnika na mestu uporabe višji, kot je maloprodajna cena števca!

Poleg ugodnosti, ki jih prinaša uporaba programske opreme v meroslovnih aplikacijah, se je treba zavedati tudi povezanih nevarnosti. Programska oprema nezadostne kakovosti je lahko vzrok veliki gospodarski škodi. Poleg tega je treba upoštevati, da je lahko programska oprema merilnih instrumentov tarča zlorab. Zato ima danes zagotavljanje kakovosti in varnosti programske opreme in merilnih podatkov imajo pomembno vlogo v meroslovju [67].

2.1 Meroslovne aplikacije, ki jih omogoča Internet [33]

Vlogo Interneta v meroslovju običajno vidimo kot internetne kalibracije ali spletne strani namenjene preskušanju programske opreme, vendar sodobna informacijska tehnologija omogoča dosti več meroslovnih aplikacij. Pridobitve razširjene uporabe Interneta na področju meroslovja lahko razdelimo na tri skupine. Prva je znatno povečana uporaba oddaljenih merilnih sistemov, npr. povezanih v porazdeljenih merilnih sistemih zakonskih meril, ali daljinsko upravljanje merilnih instrumentov v neugodnih okoljskih pogojih. Naslednje pomembno izboljšanje izhaja iz uvedbe novih meroslovnih storitev, kot so internetne kalibracije in spletna orodja za validacijo programske opreme. Nenazadnje je zelo pomembna tudi dostopnost različnih vrst meroslovnih podatkov.

Informacijska tehnologija, uporabljena v meroslovnih aplikacijah, zajema več kot javna komunikacijska omrežja, ki jih omogoča vrsta medijev, protokolov in operaterjev: optični kabel, kabelska TV omrežja, brezžična, klasična in mobilna telefonija (GSM[2], GPRS[3], UMTS[4]) ali javna omrežja z uporabo varnih kanalov (npr. VPN[5]). V porazdeljenih merilnih sistemih, zlasti v primerih, ko so distribucijska podjetja lastniki omrežij (npr. električne energije), so pogosti načini izmenjave podatkov PLC (Power Line Carrier), DLC (Distribution Line Carrier), radian (Radio Application Network) in ZigBee (IEEE 802.15.4:2006). Uporabljeni so lahko splošni internetni protokoli (SMTP[6], HTTP[7], FTP[8] ...) ali posebni protokoli, npr. DLMS (Device Language Message Specification) [15]. Z vidika varnosti informacij, uporabljajo meroslovne aplikacije standardne pristope (npr. HTTPS[9], FTPS[10] in infrastrukturo javnega ključa), po večini za varovanje podatkov (za zagotovitev pravilnosti merilnih rezultatov) in za preverjanje pristnosti udeleženih strani.

Ti pristopi niso specifični izključno za področje meroslovja, vendar jih je pomembno omeniti, da bi se metrologi zavedali njihove prisotnosti. V ozadju je zelo pogosto kot centralna točka porazdeljenega merilnega sistema postavljena podatkovna baza. Aplikacije se razlikujejo: od zbiranja podatkov, obračuna, dinamičnega tarifiranja v zakonskem meroslovju, medicinske diagnostike (podatkovne baze

[2] Global System for Mobile communications

[3] General Packet Radio Service

[4] Universal Mobile Telecommunications System

[5] Virtual Private Network

[6] Simple Mail Transfer Protocol

[7] Hypertext Transfer Protocol

[8] File Transfer Protocol

[9] Hypertext Transfer Protocol Secure

[10] File Transfer Protocol Secure

posnetih vzorcev zdravstvenih stanj) do spremljanja in upravljanja distribuiranih meroslovnih sistemov in splošnih informacij o meroslovnih zmogljivostih (npr. o kalibracijskih laboratorijih na nivoju države ali podatkov o organih, ki opravljajo kontrolo merilnih instrumentov).

2.1.1 Internetno podprte kalibracije

Pri klasičnem postopku kalibracije je bilo vedno potrebno transportirati kalibrirani instrument v laboratorij višjega reda. Tak pristop ima številne slabosti, kot so:

- kalibrirani instrument je za uporabnika dolgo neuporaben,
- stroški prevoza,
- instrument se kalibrira v pogojih, ki so različni od pogojev redne uporabe (npr. drugo osebje, povezava z drugimi instrumenti, klimatsko okolje),
- nevarnost poškodb med prevozom.

Pri kalibracijah, ki so podprte z Internetom kalibrirani instrument ostaja v svojem okolju. Posamezne tehnike se med seboj razlikujejo glede na to, ali so za njihovo izvedbo potrebni dodatni referenčni instrumenti, opredmetene mere ali materiali.

Realizacija kalibracije časa in frekvence po "Common view" [60] metodi je primer kalibracije, pri kateri za kalibracijo merilnega instrumenta zadostuje programska oprema in povezava v Internet in GPS sistem. Druge izvedenke so:

- Internetno podprte kalibracije s prenosom merilnih signalov v referenčni laboratorij [62]. V tem primeru etalonski instrument v laboratoriju višjega reda sprejema merilni signal iz laboratorija, v katerem se nahaja opredmetena mera (artefakt) na kalibraciji in na podlagi prejetega signala in znanih pogojev postavitve kalibriranega artefakta izračuna kalibracijske parametre.
- Internetno podprte kalibracije z lokalnim referenčnim standardom [61]. Pri tej izvedenki ima laboratorij – imetnik kalibriranega instrumenta lokalno referenčno opredmeteno mero znanih in zelo stabilnih lastnosti. Z instrumentom na kalibraciji izvaja meritve, jih sproti pošilja laboratoriju višjega reda in dobi nazaj kalibracijske parametre za svoj instrument.
- Internetno podprte kalibracije s potujočim etalonom. V laboratorij, v katerem je instrument katerega želimo kalibrirati, laboratorij višjega reda pošlje svoj referenčni instrument /referenčni material (potujoči etalon) katerega lastnosti pozna in redno preverja. Z instrumentom na kalibraciji se izvajajo meritve lastnosti potujočega etalona, sproti pošiljajo v laboratorij višjega reda, kjer se izračunajo parametri kalibracije in pošljejo nazaj.

Spletna orodja za validacijo programske opreme so naslednja pomembna pridobitev, ki omogoča razvijalcem meroslovne programske opreme dostop do spletnih referenčnih validacijskih orodij.

Nenazadnje je zelo pomembno to, da je na spletu dostopna ogromna količina najrazličnejših meroslovnih informacij – o meroslovnih zmogljivostih (za kalibracije in kontrolo instrumentov), navodila za metrološke postopke, tehniški standardi, zakonodaja in številne druge informacije.

3 Preskušanje programske opreme [28]

Tretje poglavje podaja pregled preskušanja programske opreme, in sicer zadev pomembnih z vidika meroslovne programske opreme. Preskušanje programske opreme je najpomembnejša komponenta validacije programske opreme, ki se izvaja v sklopu ugotavljanja njene skladnosti . V vsaki fazi razvoja programske opreme se izvajajo določene preskuševalne aktivnosti. Te faze so: analiza zahtev uporabnika, specifikacija funkcionalnih zahtev, specifikacija tehničnih zahtev, načrtovanje programskih modulov in kodiranje. Pripadajoče preskuševalne aktivnost so: preskušanje modulov, preskušanje integracije, preskušanje sistema in prevzemno preskušanje. Za uspešno vodenje in izvajanje preskušanja je pomembno razumevanje osnovnih lastnosti in pristopov k preskušanju programske opreme: obsega, načel, metod, strategij in organizacijskih vprašanj. Zaradi praktičnih razlogov v praksi ni možno doseči popolnega preskušanja. Glede na to je pomembno izbrati pravi kriterij za zaključek preskušanja, zavedajoč se, da bo v programu vedno ostalo nekaj napak. Pomembno je poudariti, da večina napak v kodi programa ostane zaradi pomanjkanja časa za njihovo odpravljanje (zaradi pritiskov po čimprejšnji izdaji proizvoda, pred konkurenco) in ne zaradi pomanjkljivega znanja ali zlih namenov programerja. Obstajajo tudi orodja za avtomatizacijo preskušanja. Njihova uporabnost je omejena in zahteva podrobno analizo od primera do primera. V nadaljevanju je podana kratka razlaga preskušanja varnosti ("security") primerna za meroslovno programsko opremo, posebnosti preskušanja internetnih aplikacij in ocenjevanje kakovosti procesa razvoja programske opreme.

4 Zahteve kakovosti za meroslovno programsko opremo [37]

Zahteve kakovosti za meroslovno programsko opremo so predstavljene v četrtem poglavju. Več vodil, ki so jih izdale različne institucije, obravnava programsko opremo meroslovnih aplikacij. Analiza teh vodil je pokazala, da nobeno od vodil ni bilo primerno za validacijo programske opreme v merilnih instrumentih MID.

Na nekaterih ključnih industrijskih področjih (npr. avtomobilski in železniški industriji, industriji vesoljskih plovil, vojaške tehnologije, opreme za nuklearne elektrarne ter biomedicinskih naprav) izvirajo zahteve za programsko opremo iz ocene posledic tveganja odpovedi programske opreme. Na teh področjih so že zdavnaj razviti in izpopolnjeni standardi, ki najpogosteje obravnavajo varnost in zanesljivost programske opreme. V primerjavi s temi področji odpovedi programske opreme zakonskih meril mogoče nimajo takojšnjih katastrofalnih posledic (škode ali poškodb). Če pa upoštevamo hierarhično strukturo nacionalnega meroslovnega sistema, lahko stopnjevalni faktor pripelje do ogromne škode (zgovoren primer je lahko nevarnost za zdravje množice ljudi, v primeru napačne kalibracije cele populacije biomedicinskih inštrumentov zaradi netočnega nacionalnega etalona).

Ciljne skupine uporabnikov, ki so zainteresirane za kakovost meroslovne programske opreme sestavljajo najmanj:

- uporabniki merilnih instrumentov,
- proizvajalci merilnih instrumentov in sistemov,
- državni organi,
- organi, odgovorni za državni sistem ugotavljanja skladnosti,
- laboratoriji, ki ugotavljajo skladnost in
- inšpekcijski organi.

Zahteve za kakovost meroslovne programske opreme izhajajo iz različnih strokovnih vidikov. To so:

- tehnični standardi, ki obravnavajo programsko opremo,
- zakonodaja,
- varnostni ("safety") tehnični standardi,
- tehnični standardi, ki obravnavajo usposobljenost laboratorijev,
- standardi varnosti ("security") programske opreme.

Ugotavljanje skladnosti meroslovne programske opreme je podprto z več tehničnimi standardi, direktivami in normativnimi dokumenti [37].

Osnovne značilnosti kakovosti programske opreme so določene v mednarodnem standardu ISO/IEC 9126-1:2001 Software engineering - Product quality - Quality model [23]. Te značilnosti so funkcionalnost, zanesljivost, uporabnost, učinkovitost, vzdrževanje in prenosljivost.
Na drugi strani je primer normativnega dokumenta, ki obravnava meroslovno programsko opremo, Evropska direktiva o merilnih instrumentih (MID) [4], ki pokriva 10 skupin merilnih instrumentov.
Primeri bistvenih zahtev za programsko opremo zakonskih meril v dodatku 1 MID so:

Člen/Zahteva

7.6. Primernost

Merilo mora biti zasnovano tako, da omogoča nadzor nad merilnimi nalogami po dajanju v promet in začetku uporabe. Posebna oprema ali programska oprema za ta nadzor mora biti po potrebi del tega merila. Preskusni postopek mora biti opisan v navodilih za uporabo.

Če merilo vsebuje programsko opremo, ki poleg merilnih zagotavlja tudi druge funkcije, mora biti programska oprema, ki je bistvena za meroslovne lastnosti, prepoznavna in nanjo ostala programska oprema ne sme nedopustno vplivati.

8.1. Zaščita pred zlorabami

Na meroslovne lastnosti merila se ne sme nedopustno vplivati s priključevanjem druge naprave, z nobeno lastnostjo priključene naprave ali s katero koli drugo oddaljeno napravo, ki komunicira z merilom.

8.3.

Programska oprema, ki je bistvenega pomena za meroslovne lastnosti, mora biti kot taka prepoznavna in zaščitena.

Merilo mora zagotoviti, da je programska oprema zlahka prepoznavna.

Dokaz o posegu mora biti na voljo razumno časovno obdobje.

8.4.

Merilni podatki, programska oprema, ki je bistvenega pomena za meroslovne lastnosti, in meroslovno pomembni parametri, ki so shranjeni ali se prenašajo, morajo biti primerno zaščiteni proti namerni ali nenamerni zlorabi.

V večini primerov so zahteve za programsko opremo vsebovane v funkcionalnih zahtevah za merilne instrumente.

Analizirani so bili naslednji dokumenti: Direktiva o merilnih instrumentih 2004/22/EC [4], standard ISO/IEC 17025 [21], vodilo WELMEC 2.3 [39], vodilo WELMEC 7.2 (zaradi primerjanja pristopov) [41], vodilo OIML D-31 General Requirements for Software-Controlled Measuring Instruments [30], EUROLAB TR 2/2006 [11], NORDTEST TR 535: Method of Software Validation [29], Terms and Conditions for the Approval of Metrological Software, Measurement Canada [27], »General Principles of Software Validation«; Final Guidance for Industry and FDA Staff: 2002 [38], FDA 21 CFR Part [12] in EN 61508 Standard for Safety-Related Systems [9].

Čeprav so pripravljena brez sistematičnih skupnih izhodišč, vsa vodila sledijo dvema pristopoma. Prvi pristop je namenjen metrologom, ki sami razvijajo meroslovne programske aplikacije in zato potrebujejo več znanja o fazah razvoja in preskušanja programske opreme. Drugi je namenjen uporabnikom, ki nabavijo že validirano aplikacijo in organom za ugotavljanje skladnosti.

5 Izdelava vodila WELMEC 7.2:2005 - Software Guide [7]

Začetno raziskovalno delo za temeljni predmet te naloge, postopek za validacijo meroslovne programske opreme, je bilo izvedeno v fazi razvoja vodila WELMEC 7.2.

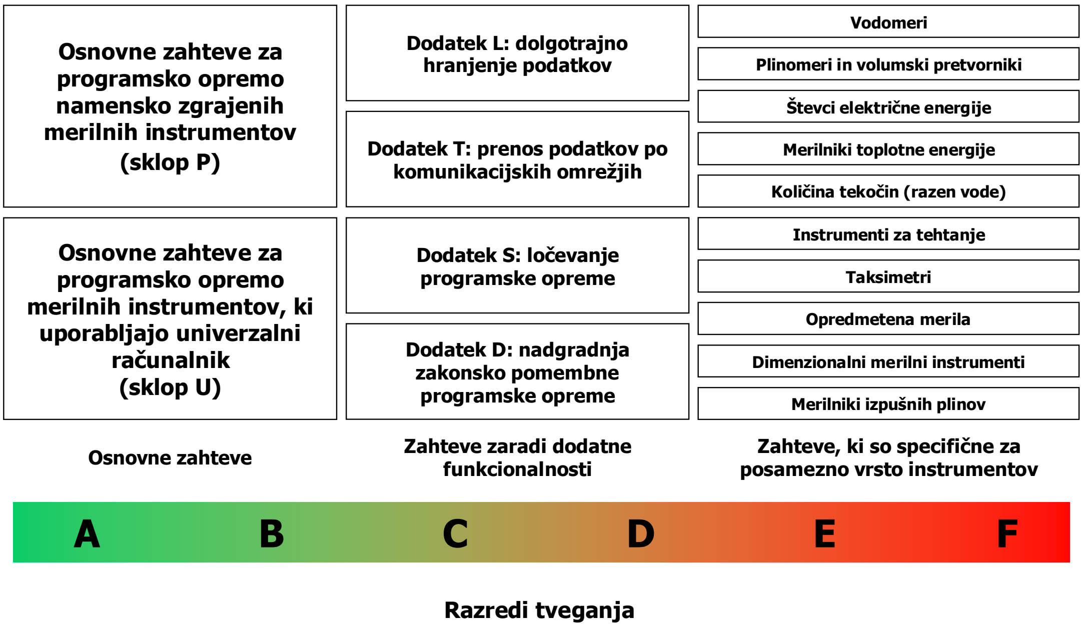
Ugotavljanje skladnosti merilnih instrumentov tradicionalno izvajajo strokovnjaki iz področja meroslovja, ki praviloma niso specialisti za preskušanje programske opreme. Mnenje na začetku projekta je bilo, da je priprava ustreznih postopkov preskušanja in izbira ustreznih metod in strategij prezahtevna naloga za njih. Na prvi pogled se je zdelo, da je rešitev tega problema prepuščanje validacije meroslovne programske opreme strokovnjakom preskušanja programske opreme. Po drugi strani je bilo jasno, da slednji nimajo temeljnih meroslovnih znanj in ne poznajo področja uporabe merilnih instrumentov. To je ocenjeno kot znatna pomanjkljivost za kakovostno preskušanje meroslovne programske opreme - rezultat takega preskušanja bi bil zelo slab glede najpomembnejših, meroslovnih vidikov preskušane programske opreme.

Po podrobni analizi v projektni skupini se je izkazalo, da je najboljša rešitev pripraviti navodila za poznavalce področja – metrologe, z nedvoumnimi navodili za preskušanje meroslovne programske opreme.

Vodilo je sestavljeno iz dveh osnovnih sklopov zahtev, ki jih je treba uporabljati alternativno: en sklop je namenjen za namensko zgrajene merilne instrumente (sklop P), drugega je pa treba uporabljati na bolj kompleksnih sistemih, ki vsebujejo univerzalni računalnik (sklop U). Glede na posebne značilnosti, ki so tipične za današnje merilne instrumente, je te osnovne zahteve treba dopolniti z dodatnimi zahtevami, in sicer z:

- zahtevami za dolgotrajno hranjenje podatkov (L: long term storage of measurement data),
- varen prenos podatkov po komunikacijskih omrežjih (T: transmission of mesurement data via communication networks),
- ločevanje zakonsko pomembne od druge programske opreme (S: software separtion) in
- zahtevami za nadgradnjo programske opreme (D: download of legally relevant software).

Poleg tega splošnega dela, ki sloni na tehnološki realizaciji računalniškega dela merilnega instrumenta, vsebuje vodilo dodatne programske zahteve, ki so specifične za posamezno vrsto merilnih instrumentov zajetih z MID. Zgradba vodila je prikazana na sliki 1.


Osnovne zahteve za programsko opremo namensko zgrajenih merilnih instrumentov (sklop P)
Osnovne zahteve za programsko opremo merilnih instrumentov, ki uporabljajo univerzalni računalnik (sklop U)
Dodatek L: dolgotrajno hranjenje podatkov
Dodatek T: prenos podatkov po komunikacijskih omrežjih
Dodatek S: ločevanje programske opreme
Dodatek D: nadgradnja zakonsko pomembne programske opreme
Vodomeri
Plinomeri in volumski pretvorniki
Števci električne energije
Merilniki toplotne energije
Količina tekočin (razen vode)
Instrumenti za tehtanje
Taksimetri
Opredmetena merila
Dimenzionalni merilni instrumenti
Merilniki izpušnih plinov
Osnovne zahteve
Zahteve zaradi dodatne funkcionalnosti
Zahteve, ki so specifične za posamezno vrsto instrumentov
A
B
C
D
E
F
Razredi tveganja

Slika 1: Zgradba vodila WELMEC 7.2

V praksi se razvijalec ali preskuševalec odloči za izbiro posameznih modulov glede na zgradbo merilnega instrumenta.

Poleg tehničnih vidikov vodilo 7.2 upošteva specifike področja uporabe merilnih instrumentov, ki jih opredeli kot razred tveganja ("risk class"). Upošteva se tveganje nedovoljenih posegov, tveganje, ki bi ga lahko povzročil premalo podroben pregled ob pregledu tipa merila in zahtevana stopnja skladnosti s programsko opremo ob odobritvi tipa.

V vsaki zahtevi je opredeljena potrebna dokumentacija in predlog validacijske metode, glede na določeni razred tveganja.

6 Verifikacija medsebojnih relacije zahtev in preskusnih metod vodila WELMEC 7.2 z uveljavljeno prakso

Za potrditev novo razvitega vodila za validacijo programske opreme je bilo nujno preveriti pravilno relacijo z vsemi povezanimi področji, predvsem zaradi zagotavljanja, da določila WELMEC 7.2 niso v nasprotju z tehniškimi standardi, normativnimi dokumenti ali vodili s področja meroslovne

programske opreme in sorodnih področij. To bi se lahko zgodilo, glede na dejstvo, da dotlej ni bilo sistematičnega pristopa in so obstoječi pristopi bili nepovezani. Preverjene so naslednje povezave:
- z zahtevami meroslovnega področja na primeru direktive o merilnih instrumentih,
- z zahtevami normativnih dokumentov in vodil s področja meroslovja (npr. WELMEC in OIML vodila),
- s tehniškimi standardi kakovosti programske opreme na primeru ISO/IEC 9126,
- s tehniškimi standardi preskušanja programske opreme ISO / IEC 12119 [17] in
- s splošnimi zahtevami ugotavljanja skladnosti.

7 Primerjalna validacija programske opreme merilnega instrumenta – izvedba, rezultati in nadaljnje aktivnosti

Sedmo poglavje predstavlja izvedbo eksperimenta, v katerem je potrjena primernost razvitega znanstvenega pristopa. Validacija vodila je bila izvedena s primerjalno validacijo enakega vzorca meroslovne programske opreme, ki je potekala hkrati v šestih laboratorijih nacionalnih meroslovnih institutov Avstrije, Nemčije, Češke, Nizozemske, Poljske in Slovenije. Namen je bil dokazati, da je vodilo WELMEC 7.2 primerno svojemu namenu, oziroma ugotoviti stopnjo enakovrednosti pristopov k validaciji programske opreme v zakonskem meroslovju med različnimi nacionalnimi organi zakonskega meroslovja. Poglavje opisuje organizacijske in logistične zadeve ter predstavi analizo rezultatov in nadaljnje aktivnosti.

Dogovorjen je bil naslednji način izvajanja eksperimenta:

1. Vsak udeleženec samostojno opredeli zahteve na podlagi vodila WELMEC 7.2.
2. Validacijo je treba izvesti v skladu z navodili zapisanimi v vodilu. Izbira metod je bila prepuščena izvajalcem.
3. Predlagani postopek za opravljanje dela v laboratoriju, je:

- določanje zahtev,
- izbira metod za preverjanje zahtev,
- priprava podrobnega načrta preskusa, vključno definicijo preskusnega okolja, izbire preskusnih metod in določanjem testnih primerov,
- izvedba validacije,
- priprava poročila in pošiljanje koordinatorju eksperimenta (Urad RS za meroslovje).

4. V primeru odkritja napake med preskušanjem:

- zapisati dejansko stanje (opis napake, okolje, pogoji),
- nadaljevanje preskušanja z istim preskušancem (z neopremljeno programsko opremo).

5. Vsak udeleženec kot rezultat validacije pripravi naslednje dokumente:

- načrt preskusa (test plan),
- poročilo o preskusu (v skladu s poglavjem 12 WELMEC 7.2) ,

- odstavek o programski opremi za certifikat o odobritvi tipa (v skladu z WELMEC 7.2, točka 12.4),
- kratko poročilo o preskušanju, vključno s pripombami in zapisom nejasnosti v vodilu in predlogi izboljšanja,
- podroben pregled preskušanja z naslednjimi vsebinami:
    - identificirane zahteve,
    - uporabljene preskusne metode (za vsako posamezno zahtevo),
    - ugotovitev o izpolnjevanju zahteve (za vsako posamezno zahtevo).

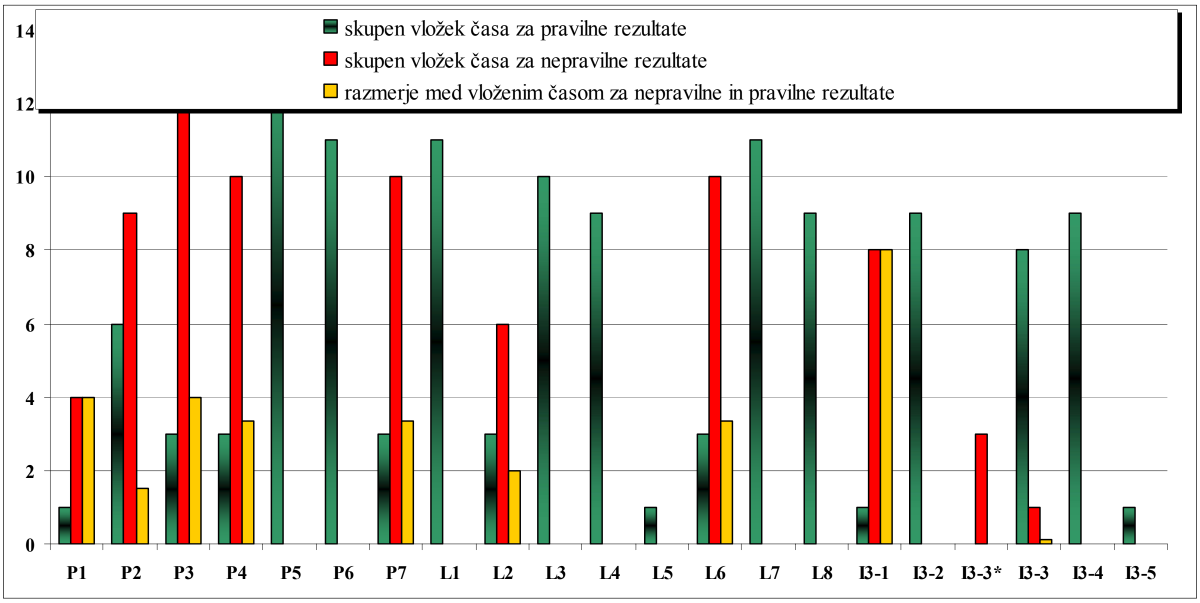
Na prvi pogled so rezultati eksperimenta bili izjemno nezadovoljivi. Le devet zahtev od enaindvajsetih je bilo od vseh sodelujočih laboratorijev ocenjenih kot izpolnjenih. Ugotovitve glede izpolnjevanja preostalih dvanajstih zahtev so bile različne in razlike niso bile enake. Izjemno zaskrbljujoče je bilo dejstvo, da so bile splošne presoje skladnosti različne: štirje sodelujoči laboratoriji so ugotovili, da je preskušana programska oprema skladna z zahtevami, dva laboratorija sta ugotovila, da ni skladna! To je povzročilo takojšnjo temeljito analizo vzrokov in pripravo korektivnih ukrepov. Zgoščen pregled rezultatov eksperimenta je predstavljen v tabeli 2. Osenčene celice predstavljajo zahteve, katerih izid (ustreza / ne ustreza) ni enak pri vseh udeležencih, ne glede na uporabljene metode preskušanja. Podrobna analiza izidov validacije posameznih zahtev po posameznih izvajalcih je dala zanimive rezultate. Kot referenčni rezultat je v analizi izbran najstrožji izid za vsako posamezno zahtevo. Poraba časa za preskušanje je utežena kot:

1 – za preskušanje na podlagi pregleda dokumentacije

2 - za funkcionalni preskus.

	A			B			C			D			E			F	
Zahteva	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**	**IDE**	**MET**	**PAS**
P1 - Dokumentacija	+	D	+	+	D	-	+	D	+	+	D	+	+	D	+	+		+
P2 - Identifikacija programske opreme	+	D,F	+	+	D,F	-	+	D,F	+	+	D,F	-	+	D,F	+	+		+
P3 - Vpliv preko uporabniškega vmesnika	+	D,F	+	+	D,F	-	+	D,F	+	+	D,F	+	+	D,F	+	+		+
P4 - Vpliv preko komunikacijskega vmesnika	+	D,F	+	+	D,F	-	+	D	+	+	D,F	+	+	D,F	+	+		+
P5 -Zaščita pred nenamernimi spremembami	+	*D,F*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D,F*	+	+		+
P6 - Zaščita pred namernimi spremembami	+	*D,F*	+	+	*D*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+		+
P7 - Zaščita parametrov	+	D	+	+	D,F	-	+	D,F	+	+	D,F	+	+	D	+	+		+
L1 - Popolnost shranjenih podatkov	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D*	+	+		+
L2 - Zaščita pred nenamernimi spremembami	+	D	+	+	D	+	+	D	+	+	D,F	-	+	D,F	+	+		+
L3 - Popolnost podatkov	+	D	+	+	D,F	+	+	D	+	+	D,F	+	N.A.		N.A.	+		+
L4 - Verodostojnost shranjenih podatkov	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D*	+	+		+
L5 - Zaupnost gesel	N.A.		N.A.	N.A.		N.A.	N.A.		N.A.	N.A.		N.A.	N.A.		N.A.	N.A.		N.A.
L6 - Vzpostavitev shranjenih podatkov	+		+	+	D,F	+	+	D	+	+	D,F	-	+	D,F	+	-		N.A.
L7 - Samodejno shranjevanje	+	*D*	+	+	*D*	+	+	*D,F*	+	+	*F*	+	+	*D,F*	+	+		+
L8 – Zmogljivost in stalnost spomina	+	*D*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+		+
I3-1 - Vzpostavitev po odpovedi	+	D,F	+	+	D	?	+	*D*	+	+	D,F	+	+	D	+	+		+
I3-2 - Zmogljivosti rezervnega spomina	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D,F*	+	+	*D*	+	+		+
I3-3 – Funkcionalnosti alarmov in ponovne vzpostavitve ???	+	D	+	N.A.		N.A.		N.A.		N.A.	.	N.A.	.	N.A.	.	N.A.	.	N.A.
I3-3 - Primernost prikaza	+	*D*	+	+	*D,F*	+	+	*D*	+	+	*D*	+	+	*D*	+	+		+
I3-4 – Preprečitev brisanja kumulativnih izmerjenih vrednosti	+	*D,F*	+	+	*D,F*	+	+	*D*	+	+	*D,F*	+	+	*D,F*	+	+		+
I3-5 – Dinamično delovanje	N.A.		N.A.	N.A.		N.A.	+	D	+	N.A.		N.A.	N.A.			N.A.		N.A.

Tabela 2: Pregled rezultatov: IDE- zahteva je bila identificirana; MET: metoda; PAS: ustreznost (ustreza " +")/(ne ustreza "-"); D: analiza dokumentacije; F: funkcionalni test, N.A. ni primerno



Slika 2: pregled porabljenega časa za preskušanje posameznih zahtev, glede na pravilnost rezultata

Podatki, predstavljeni sliki 2, najbolje ilustrirajo razumevanje zahtev. Na sliki so predstavljeni:

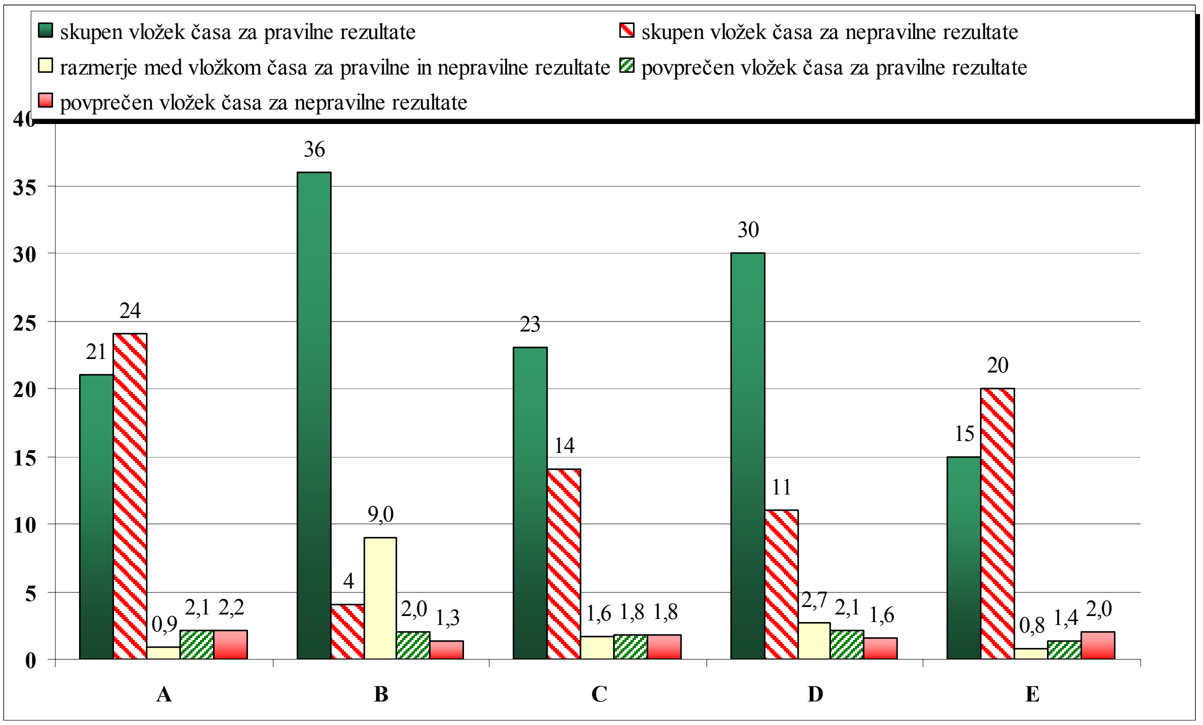
- skupen porabljen čas vseh udeležencev v eksperimentu za validacijo posamezne zahteve s pravilnim izidom,
- skupen porabljen čas vseh udeležencev v eksperimentu za validacijo posamezne zahteve z napačnim izidom,
- razmerje med porabljenim časom validacije posamezne zahteve z napačnim in porabljenim časom validacije s pravilnim izidom.

Če pogledamo razlike med referenčnim rezultatom in rezultatom validacije za vsako posamezno zahtevo, je očitno, da so najbolje razumljene zahteve: P5, P6, L1, L4, L5, L7, L8, I3-2 in I3-4. Najslabše razumljene zahteve so: P1, P3, P4, P7, L2, L6 in I3-1. Logičen zaključek je bil, da je te zahteve v vodilu WELEMC 7.2 potrebno dodatno razložiti.

Na sliki 3 je predstavljen en pogled na učinkovitost izvajanja preskušanja v posameznih laboratorijih, in sicer glede porazdelitve dela, ki pripelje do točnih ugotovitev in dela, ki pripelje do napačnih ugotovitev. Na sliki 3 so predstavljeni:

- skupen porabljen čas za validacijo vseh zahtev s pravilnim izidom, po udeležencu,
- skupen porabljen čas za validacijo vseh zahtev z napačnim izidom, po udeležencu,
- razmerje med skupnim porabljenim časom za pravilne in napačne izide,
- povprečno porabljen čas za pravilen izid, po udeležencu,
- povprečno porabljen čas za napačen izid, po udeležencu.

Če pogledamo razmerje med skupnim porabljenim časom za pravilne in napačne izide, lahko rečemo, da je udeleženec "B" 11.25 krat učinkovitejši kot udeleženec "E". To je pomemben kazalec potrebnih izboljšav dela laboratorija.



Slika 3: pregled porabljenega časa za preskušanje po udeležencih, glede na pravilnost rezultata

Nadaljnja analiza je pokazala, da je glavni vzrok različnih ocen izpolnjevanja posameznih zahtev v njihovem različnem razumevanju. Poglaviten razlog za to je bil v neustrezni opredelitvi zahtev za zaščito posameznih kategorij parametrov v spominu števca.

Čeprav to ni bil prvotni namen eksperimenta, analiza rezultatov je pokazala, da:

- obseg dela (vloženi čas) ni neposredno povezan s kakovostjo rezultatov preskušanja,
- izbira preskusnih metod ne vpliva direktno na kakovost rezultatov preskušanja.

Glavna vpliva na kakovost preskušanja so pravilno razumevanje zahtev in usposobljenost (izkušnje) izvajalca.

8 Prispevki k znanosti

Najpomembnejše delo opisano v pričujoči disertaciji obsega:

- Celovito analizo in vrednotenje obstoječih vodil glede na njihovo uporabnost za zakonska merila.
- Oblikovanje vodila za validacijo programske opreme v merilnih instrumentih, reguliranih z direktivo o merilnih instrumentih.
- Potrditev, da je izdelano vodilo skladno z obstoječo prakso povezanih področij. To je še posebej pomembno v izrazito interdisciplinarnih primerih. V primeru predstavljenega dela so ta področja tehnična (meroslovje, kakovost programske opreme, preskušanje programske opreme in zagotavljanje kakovosti) in družbena (zakonodaja, ugotavljanje skladnosti, zaščita potrošnikov,

prost pretok blaga in storitev). Pomembno je, da predlagana nova praksa ni v nasprotju z že uveljavljenimi pravili povezanih področij.

- Razvoj in eksperimentalno potrditev metode primerjalne validacije meroslovne programske opreme, kot metode za validacijo novo razvitega vodila.

Doseženi so bili naslednji prispevki k znanosti:

8.1 Izbira primernih zahtev, pristopov, postopkov in tehnik iz obstoječih dokumentov za programsko opremo

Ta analiza je bila pomembna za potrditev, da je razvito vodilo, ki izhaja iz strokovnega področja zakonskega meroslovja, skladno s stanjem tehnike na področju kakovosti in preskušanja programske opreme. Ugotovljene šibke točke dodatno potrjujejo izbrano metodologijo:

- WELMEC 7.2 ne pokriva v zadostni meri lastnosti kakovosti uporabnosti ("usability"),
- WELMEC 7,2 obravnava varnost ("security") veliko bolj kot ISO/IEC 12119, posebej funkcionalne preskuse varnosti,
- morebitne odpovedi programske opreme se obravnavajo bolj z vidika funkcionalnosti kot z vidika zanesljivosti.

8.2 Vzpostavitev povezav med področnimi standardi za komponente programske opreme, generičnimi standardi kakovosti programske opreme, standardi preskušanja programske opreme ter uporabljenimi strategijami preskušanja programske opreme

Pomembnost te analize je v tem, da dokazuje povezanost med tehnikami validacije programske opreme na posameznem področju (npr. v merilnih instrumentih) in običajnimi postopki in strategijami v praksi preskušanja programske opreme. Povezava teh dveh skupin zagotavlja, da validacija programske opreme z izhodiščem v strokovnem področju (meroslovja) izpolnjuje bistvene strokovne zahteve kakovosti in preskušanja programske opreme. Dodatno so bili pri tem identificirani najprimernejši postopki validacije programske opreme za strokovnjake z drugih področij.

8.3 Razvoj generičnega pristopa k validaciji meroslovne programske opreme

To vprašanje je pomembno zaradi dejstva, da obstaja več področij, kjer programska oprema igra pomembno vlogo, hkrati pa strokovnjaki iz teh področij ne obvladujejo validacije programske opreme. Namen je bil dokazati, da je možno pripraviti ustrezna navodila za validacijo programske opreme za strokovnjake iz področij na katerih se ta programska oprema uporablja. Vodilo WELMEC 7.2, kot rezultat skupinskega dela, je primer uspešne aplikacije takšnega generičnega pristopa na področju validacije programske opreme merilnih instrumentov. Vodilo pokriva celoten proces validacije programske opreme zakonskih meril, ki je sestavljen iz naslednjih faz:

- prepoznavanje bistvenih zahtev programske opreme in IT zahteve za posamezne skupine merilnih instrumentov, glede na njihovo tehnološko izvedbo,
- opredelitev gradnikov modularne strukture,
- prečiščevanje teh zahtev na zahteve primerne za preskušanje,
- navodila za validacijo teh zahtev za strokovnjake iz drugih področij,
- navodila za usklajeno poročanje o rezultatih in ugotovitvah.

Vodilo ni uporabno samo za merilne instrumente regulirane z MID – ta koncept je primeren tudi za merilne instrumente, ki se uporabljajo na nereguliranih področjih.

8.4 Razvoj in validacija postopka ter primerjalno preskušanje enega izdelka programske opreme na enem instrumentu

Pomen te teme izhaja iz potrebe po zagotavljanju primernega načina validacije novo razvitega vodila (predstandarda), za validacijo meroslovne programske opreme. Takšna metodologija (primerjalna validacija iste komponente programske opreme, ki jo izvaja več institucij) še ni bila izvedena na tem tehničnem področju.

Izvedba eksperimenta je bila bistvenega pomena za potrditev primernosti predlagane metode primerjalne validacije programske opreme. Na prvi pogled so bile ugotovitve o skladnosti posameznih zahtev zelo različne (udeleženci so prišli do enakih sklepov o izpolnjevanju le 9 od skupaj 21 zahtev). Štirje od šestih udeležencev v eksperimentu so zmotno ugotovili, da je preskušana programska oprema skladna z zahtevami. Nadaljnja analiza pa je pokazala, da je glavni razlog za te razlike v različnem razumevanja posameznih zahtev WELMEC 7.2. Popravki so že bili upoštevani v tretji izdaji vodila iz leta 2008.

Zanimive so bile tudi ugotovitve glede povezave med uporabljeno metodo preskušanja in rezultatom preskusa: ni bilo ugotovljene neposredne povezave. Prav tako ni bilo korelacije med vloženim naporom v preskušanje posamezne zahteve in pravilnostjo ugotovitve o skladnosti. Pokazalo se je, da ima izkušenost preskuševalca zelo velik pomen, vendar le-te ni možno neposredno izmeriti. Čeprav namen eksperimenta ni bil ocenjevanje dela posameznih laboratorijev, se nekatere ugotovitve lahko uporabijo v ta namen.

Organizacija eksperimenta je bila ustrezna, izvedba je zagotovila pravočasno informacijo o možnostih harmonizirane implementacije direktive o merilnih instrumentih na področju programske opreme merilnih instrumentov.

Razvita metodologija primerjalne validacije programske opreme ima najmanj tri možne aplikacije:

- validacija novih vodil (standardov),
- določanje nivoja enakovrednosti laboratorijev, ki izvajajo validacijo programske opreme (identifikacija pomanjkljivosti),
- koristno orodje za vzdrževanje enkrat dosežene zadovoljive stopnje enakovrednosti (periodična preverjanja sodelujočih laboratorijev).

9 Zaključki

Validacija s primerjalnim preskušanjem je pokazala, da je vodilo WELMEC 7.2 primerno svojemu namenu – da omogoči primerljiv, harmoniziran pristop k ugotavljanju skladnosti na področju programske opreme merilnih instrumentov, ki so zajeti z Evropsko direktivo o merilnih instrumentih. Poleg tega se je izkazalo, da je metoda primerjalnega preskušanja programske opreme učinkovito orodje z več vidikov. Prvi vidik je uporabnost za validacijo novih podobnih vodil. Naslednji zelo pomemben vidik je ugotavljanje stopnje ekvivalentnosti implementacije vodila v praksi. Zelo pomembna je tudi možnost uporabe primerjalnega preskušanja za vzdrževanje enkrat dosežene stopnje ekvivalentnosti in izboljševanje preskuševalnih postopkov v sodelujočih laboratorijih.

Na podlagi analize rezultatov primerjalnega preskušanja so bili narejeni potrebni popravki vodila in nekateri napotki, pomembni za skupno, enotno razumevanje zakonskega meroslovja v evropskem prostoru in širše. Izvedba podobnih eksperimentov bo v prihodnosti zelo koristna za izboljševanje postopkov preskušanja programske opreme. Predmet predstavljenega primerjalnega preskušanja je bil merilni instrument preproste tehnološke izvedbe in ugotavljanje skladnosti po modulu »B« (pregled tipa). Zelo dobro bi bilo validirati dele vodila, ki obravnavajo zahtevnejše konfiguracije merilnih instrumentov s stališča programske opreme in informacijske tehnologije (ki temeljijo na univerzalnih računalnikih; v sistemih, kjer se merilni rezultati prenašajo po različnih komunikacijskih omrežjih ali pri merilih z možnostjo oddaljenega nalaganja programske opreme). Bodoči razvoj bo zelo verjetno razširitev vodila na druge module ugotavljanja skladnosti (npr. H1 "izjava o skladnosti na podlagi celovitega zagotavljanja kakovosti in pregleda zasnove", ki je zelo zanimiv za proizvajalce merilnih instrumentov). Primerjanje pristopov k modulu H1 bo zelo pomembno za določanje smeri razvoja zakonskega meroslovja. Nenazadnje je primerjalno preskušanje programske opreme lahko učinkovito orodje tudi za druga področja in ne samo za zakonsko meroslovje.

Primerjalno preskušanje programske opreme, ki je predstavljeno v tej disertaciji, do sedaj še ni bilo izvedeno na področju meroslovne programske opreme. Posledično o tem področju v svetovnih merilih ni bilo sistematičnega znanja v času začetka izvajanja eksperimenta. Namen eksperimenta ni bil ugotavljanje usposobljenosti posameznih udeležencev, temveč validacija izdelanega vodila za preskušanje meroslovne programske opreme, pridobivanje znanja o možnosti harmonizirane implementacije obstoječih pristopov k preskušanju programske opreme in preverjanje uporabnosti metode primerjalnega preskušanja programske opreme nasploh.

Najpomembnejši izrazi, ki so uporabljeni v disertaciji so razloženi v poglavju 10.8.

List of literature

1. BIPM Guidelines for CIPM key comparisons, 2005.

2. BS 7925 - Software Component Testing, 1998.

3. Directive 90/384/EEC on non automatic weighing instruments, 1990.

4. Directive on measuring instruments (2004/22/EC), (MID), 2004.

5. Drnovšek, J.: Tehniški standardi s področja kakovosti na različnih področjih v povezavi z merilno tehniko. Podgorica: Seminar Izkušnje in novosti pri uporabi standarda SIST EN ISO/IEC 17025, 15.4. 2004.

6. EN 50470-1:2006 CLC/TC 13 Electricity metering equipment (A.C.) -- Part 1: General requirements, tests and test conditions - Metering equipment (class indexes A, B and C) 6060 2004/108/EC, 2004/22/EC.

7. Richter, D., Grottker, U., Talebi, D., Schwartz, R.: The new European software guide for legal metrology: Basic principles, Computer Standards & Interfaces 28, 2006, pages 270-276.

8. EN 50470-3:2006 CLC/TC 13 Electricity metering equipment (A.C.) -- Part 3: Particular requirements - Static meters for active energy (class indexes A, B and C) 6060 2004/22/EC.

9. EN 61508, Functional safety of electrical / electronic / programmable electronic safety-related systems, Parts 1 – 7, 2002-2003.

10. EN ISO/IEC 17000:2004 Conformity assessment – Vocabulary and general principles.

11. EUROLAB TR 2/2006: Guideline for the use of computers and software in laboratories with reference to ISO 17025, October 2006.

12. FDA: Guidance for Industry Part 11, Electronic Records; Electronic Signatures — Scope and Application, 2003.

13. Greif, N.: Backtracing of Software Requirements, MID Software, WP3, Final Report, 2003.

14. Greif, N.: MID Module H1 and Software Processes, WELMEC WG7 12th meeting, 08/09 March 2007, Berlin.

15. IEC 62056-21:2002 DLMS Device Language Message specification (http://www.dlms.com).

16. International vocabulary of terms in legal metrology:2000 (VIML).

17. ISO/IEC 12119:1994, Software Packages – Quality Requirements and Testing.

18. ISO/IEC 12207: 1995: Information Technology -Software Life Cycle Processes.

19. ISO/IEC 15408: 1999: Common Criteria for Information Technology Security Evaluation (CC, version 2.1).

20. ISO/IEC 15504: 2004: Software Engineering -Process Assessment (SPICE).

21. ISO/IEC 17025:2005, General requirements for the competence of testing and calibration laboratories.

22. Hermann, Debra S: Software Safety and Reliability, IEEE Computer Society Press, 1999, ISBN 0769502997.

23. ISO/IEC 9126-1:2001, Software Engineering – Product Quality – Part 1: Quality Model.

24. Jäger F., Grottker U., Schrepf H., Guse W.: Protection of image and measurement data in an open network for traffic enforcement, Computer Standards & Interfaces 28, 2006, pages 311– 326.

25. ISO/IEC Guide 43; 1997: Proficiency testing by interlaboratory comparisons:
    Part 1: Development and operation of proficiency testing schemes,
    Part 2: Selection and use of proficiency testing schemes by laboratory accreditation bodies.

26. Jacobson, J.: Validation of software in measuring instruments, Computer Standards & Interfaces 28, 2006, pages 277–285.

27. Terms and Conditions for the Approval of Metrological Software, Measurement Canada, 2009.

28. Myers, J. G.: The Art of Software Testing, Second edition; John Wiley & Sons, Inc., 2004; ISBN 0-471-46912-2.

29. NORDTEST TR 535: Method of Software Validation, 2003.

30. OIML D-31:2008 General Requirements for Software Controlled Measuring Instruments.

31. OIML R 120: 1996 Standard capacity measures for testing measuring systems for liquids other than water.

32. Tasić T.: Validation of the software for automation of measurements, master degre thesis, University in Ljubljana, Faculty of electrical engineering, 2001.

33. Pavese, F., Forbes, A.B. (Eds.): Data Modelling for Metrology and Testing in Measurement Science, Springer-Birkhauser, 2009, ISBN 978-0-8176-4592-2.

34. R 21 – EN: 2007, Taximeters: Metrological and technical requirements, test procedures and test report format.

35. R 49-1 – EN: 2006, Water meters intended for the metering of cold potable water and hot water. Part 1: Metrological and technical requirements.

36. Stolz, H.: What is the difference between B+F or B+D and H1?, WELMEC WG7 12th meeting, 08/09 March 2007, Berlin.

37. Tasić T. and Grottker U.: An overview of guidance documents for software in metrological applications. Computer Standards & Interfaces: Special Issue on Validation of Software in Metrology, 28, 2006, pages 256-269.

38. U.S. Department Of Health and Human Services, Food and Drug Administration: General Principles of Software Validation; Final Guidance for Industry and FDA Staff, January 11, 2002.

39. WELMEC 2.3: Guide for Examining Software (Non-automatic Weighing Instruments), 1995.

40. WELMEC 2.5: Guide for modular approach and testing of PCs and other digital peripheral devices (Non-automatic Weighing Instruments), 2000.

41. WELMEC 7.2:2009 - Software Guide (Measuring Instruments Directive 2004/22/EC).

42. WELMEC 8.0:2007 - Measuring Instruments Directive 2004/22/EC, Generalities on the Assessment and Operation of Notified Bodies performing Conformity Assessment.

43. WELMEC 8.2:2007- Guide for Measuring Instruments Directive 2004/22/EC Application of Module H1.

44. WELMEC 8.3:2007- Measuring Instruments Directive 2004/22/EC, Application of Module B.

45. WELMEC 8.4:2007- Measuring Instruments Directive 2004/22/EC, Application of Module D.

46. WELMEC 8.5:2007- Measuring Instruments Directive 2004/22/EC, Assessment of Notified Bodies in Charge of Type Examination Presumption of Conformity based on EN 45011.

47. WELMEC 8.6:2007- Measuring Instruments Directive 2004/22/EC, Presumption of Conformity of the Quality System of Manufacturers with Module D or H 1 when EN ISO 9001:2000 is applied.

48. Tasić, T.: Report on the Workshop on Future Aspects of Software and IT in Legal Metrology (FASIT), 25–26 September 2003, Ljubljana, Slovenia, Bulletin de Organisation internationale de métrologie légale, Vol. XLV – Nr. 1, January 2004, vol. 45, no. 1, pages 49-51

49. Greif, N.: Software testing and preventive quality assurance for metrology, Computer Standards & Interfaces 28, 2006 , pages 286– 296.

50. BIPM: Bureau International des Poids et Mesures , http://www.bipm.org, 2009-06-16.

51. OIML - International Organization of Legal Metrology, http://www.oiml.org, 2009-06-16.

52. European Cooperation in Legal Metrology; WELMEC; http://www.welmec.org/, 2009-06-16.

53. WELMEC WG 7 "Software", http://www.welmecwg7.ptb.de/, 2009-06-16.

54. Greif, N. and Schrepf, H.: Data flow analysis in legal metrology 260, Ciarlini, P., Cox, M.G., Pavese, F. and Richter, D. (eds.): "Advanced Mathematical Tools in Metrology, vol.3", Series on Advances in Mathematics for Applied Sciences vol.45, World Scientific, Singapore, 1997.

55. Jiang, Z.,: Towards a TWSTFT network time, Metrologia 45, 2008, S6–S11.

56. ISO - International Organization for Standardization, http://www.iso.org, 2009-06-16.

57. IEC - International Electrotechnical Commission, http://www.iec.ch, 2009-06-16.

58. SELMA-Konsortium. Sicherer Elektronischer Messdaten-Austausch, http://www.selma-project.de/, 2009-06-16.

59. Gareth, F.: Basic concepts of internet metrology, Workshop on Internet Metrology and Self-Calibration, NMi, Delft, The Netherlands, 2002.

60. Allan, D. and Weiss, M.: Accurate time and frequency transfer common view of a GPS satellite. In Proceedings of 34th Annual Frequency Symposium, USAERADCOM, Ft. Monmouth, NJ 07703, 1980.

61. Parkin G. and Barker R. M.: SSfM Good Practice Guide No 19: Internet-enabled Metrology Systems, National Physical Laboratory, Hampton Road, Teddington, Middlesex, TW11 0LW, United Kingdom, 2006.

62. Matsumoto H., Sasaki K., and Hirai A.: Remote calibration of practical lengths by using low-coherence interferometry and optical fibre network. In PTB-BIPM Workshop on the Impact of Information Technology in Metrology, Berlin, Germany, 2007.

63. Parkin G. and Harris P.: Ensuring numerical correctness using the internet. Computer Standards & Interfaces: Special Issue on Validation of Software in Metrology, 28, 2006, pages 297–305.

64. Tasić T., Bojkovski J.: Intercomparison of metrological software modules—can it be useful for metrological laboratories? Joint BIPM–NPL Workshop on the Impact of Information Technology in Metrology, NPL, Teddington, UK, 16–19 September 2002.

65. Premuš A., Tasić T., Palmin U., Bojkovski, J.: Validation of Web application for testing of temperature software, Ciarlini, P., Cox, M.G., Pavese, F., Richter, D. and Rossi, G.B. (eds.): "Advanced Mathematical Tools in Metrology, vol.7", Series on Advances in Mathematics for Applied Sciences vol. 66, World Scientific, Singapore, 2006. ISBN: ISBN 86-435-0025-9.

66. Tasić T., Urleb M., and Grgić G.: System of databases for supporting co-ordination of processes under responsibility of Metrology Institute of Republic of Slovenia, Ciarlini, P., Cox, M.G., Pavese, F., Richter, D. and Rossi, G.B. (eds.): "Advanced Mathematical Tools in Metrology, vol.7", Series on Advances in Mathematics for Applied Sciences vol. 66, World Scientific, Singapore, 2006. ISBN: ISBN 86-435-0025-9.

67. Richter, D.: Validation of software in metrology, Computer Standards & Interfaces 28, Editorial, 2006, pages 253–255.

68. Directive 2006/95/EC of the European Parliament and of the Council of 12 December 2006 on the harmonisation of the laws of Member States relating to electrical equipment designed for use within certain voltage limits (LVD).

69. Directive 2004/108/EC of the European Parliament and of the Council of 15 December 2004 on the approximation of the laws of the Member States relating to electromagnetic compatibility and repealing Directive 89/336/EEC (EMC).

70. Directive 2002/95/EC of the European Parliament and of the Council of 27 January 2003 on the restriction of the use of certain hazardous substances in electrical and electronic equipment (RoHS).

71. 90/683/EEC: Council Decision of 13 December 1990 concerning the modules for the various phases of the conformity assessment procedures which are intended to be used in the technical harmonization directives.

72. European Association of National Metrology Institutes, http://www.euramet.org/, 2009-06-16.

73. A Focus for Analytical Chemistry in Europe, http://www.eurachem.org/, 2009-06-16.

74. European Federation of National Associations of Measurement, Testing and Analytical Laboratories, http://www.eurolab.org, 2009-06-16.

75. European co-operation for Accreditation, http://www.european-accreditation.org/, 2009-06-16.

76. Euro-Asian Cooperation of National Metrological Institutions, http://www.coomet.org/, 2009-06-16.

77. Directive 2001/16/EC Interoperability of the trans-European conventional rail system.

78. EN 50128:2001 Railway applications - Communication, signalling and processing systems - Software for railway control and protection systems.

79. Regulation (EC) No 552/2004 - Interoperability of the European Air Traffic Management network.

80. The Motor Industry Software Reliability Association: http://www.misra.org.uk/, 2009-06-13.

81. ISO/TR 15497:2000: Road vehicles - Development guidelines for vehicle based software.

82. IEC 60880: 2006: Nuclear power plants - Instrumentation and control systems important to safety - Software aspects for computer-based systems performing category A functions.

83. Directive 93/42/EEC on Medical devices.

84. IEC 60601-1-4:2007: Medical electrical equipment - Part 1-4: General requirements for safety - Collateral Standard: Programmable electrical medical systems.

85. Radio Technical Commission for Aeronautics (RTCA): 1999: DO-178B, Software Considerations in Airborne Systems and Equipment Certification.

86. European Cooperation for Space Standardisation: ECSS-Q-80B: 2003, Software product assurance.

87. Dogša, T.: Verifikacija in validacija programske opreme : V & V. 1. izd. Maribor: Tehniška fakulteta, Elektrotehnika, računalništvo in informatika, 1993, ISBN 86-435-0025-9.

88. Tasić, T., Richter, D., Grotker, U., Kok, P., and Rahm, C.: Experience gained from a comparative examination of measuring instrument software, Bulletin Organisation Internationale de Métrologie Légale, Volume XLIX, Nr. 2, April 2008, pages 23-27.

89. Tasić, T., Grotker, U., Just, S.: Preparation of the first OIML Working Document on Software in measuring instruments, Bulletin Organisation Internationale de Métrologie Légale, Volume XLVII • Nr. 2, April 2006, pages 25-29.

90. Rayner, D. and Barker, R.: METROS - A web site for algorithms for metrology and associated guidance, Ciarlini, P., Cox, M.G., Filipe, E., Pavese, F. and Richter, D. (eds.): "Advanced Mathematical Tools in Metrology, vol.5", Series on Advances in Mathematics for Applied Sciences vol. 57, World Scientific, Singapore, 2001, ISBN: 981-024.4940.

91. Hartmann, V., Gross, H. and Richter, D.: Databases in Metrology: requirements and solutions, Ciarlini, P., Cox, M.G., Filipe, E., Pavese, F. and Richter, D. (eds.): "Advanced Mathematical Tools in Metrology, vol.5", Series on Advances in Mathematics for Applied Sciences vol. 57, World Scientific, Singapore, 2001, ISBN: 981-024.4940.

92. Bojkovski, J., Drnovšek, J., Pušnik, I., Tasić, T.: Automation of a precision temperature calibration laboratory. IEEE transactions on instrumentation and measurements, 2000, Vol. 49, No. 3, pages 596-601.

93. D. W. Allan, D.W. and Thomas, C.: Technical Directives for Standardization of GPS Time Receiver Software, Metrologia, 1994, 31, pages 69-79.

94. El Emam, K. and Andreas Birk, A.: Validating the ISO/IEC 15504 Measure of Software Requirements Analysis Process Capability, IEEE Transactions On Software Engineering, Vol. 26, No. 6, June 2000, pages 541-566.

95. Yang M. C. K., Chao, A.: Reliability -Estimation & Stopping-Rules for Software Testing, Based on Repeated Appearances of Bugs, IEEE Transactions On Reliability, Vol. 44, No. 2. 1995 June, pages 315-321.

96. Basili, V. R. and Selby, R. W.: Comparing the Effectiveness of Software Testing Strategies, IEEE Transactions on Software Engineering, Vol. Se-13, No. 12, December 1987, pages 1278-1296.

97. Bertino, E. and Sandhu, R.: Database Security—Concepts, Approaches, and Challenges, IEEE Transactions on Dependable And Secure Computing, Vol. 2, No. 1, January-March 2005, pages 2-19.

98. Huber, L.: Qualification and validation of software and computer systems in laboratories Part 2: Qualification of vendors, Accreditation and Quality Assurance (1998) 3: 2–5, Q Springer-Verlag 1998, pages 371-381.

99. Redgrove, J., Filtz, J.-R., Fischer, J., Le Parlouër, P., Mathot, V., Nesvadba, P., Pavese, F.: EVITHERM: The Virtual Institute of Thermal Metrology, International Journal of Thermophysics Vol. 28, Number 6 / December, 2007, pages 2155–2163.

100. Thomas, N., Reeves, H.: experience from quality assurance in nuclear power plant protection system software validation, IEEE Transactions on Nuclear Science, Vol. NS-27, No. 1, February 1980, pages 899-908.

101. Leveson, N. G. and Harvey, P.: Analyzing Software Safety, IEEE transactions on software engineering, vol. Se-9, no. 5, September 1983, pages 569-579

102. Huand, C-H. and Lyu, M.R.: Optimal Release Time for Software Systems Considering Cost, Testing-Effort, and Test Efficiency, IEEE transactions on reliability, vol. 54, no. 4, December 2005, pages 583-591.

103. Brown, D. B. Maghsoodloo, S. and Deason, W. H.: A cost model for determining the optimal number of software test cases, IEEE transactions on software engineering. Vol. 15. No. 2. February 1989, pages 218-221.