FEDERAL STATE UNITARY ENTERPRISE
THE ALL-RUSSIAN RESEARCH INSTITUTE
OF THE METROLOGICAL SERVICE (FSUE "VNIIMS")

RECOMMENDATION

STATE SYSTEM FOR ENSURING UNIFORM MEASURMENT

GENERAL SOFTWARE REQUIREMENTS FOR MEASURING INSTRUMENTS

MI 2891 - 2004

Moscow
2004

FORWORD

DEVELOPED by FSUE "VNIIMS" of the Federal Agency on Technical Regulation and Metrology

EXECUTIVES U.E. Lukashov, Ph.D. in technical science (project head), U.A. Kudeyarov, Ph.D. in physical and mathematical science, A.A. Satanovskiy

APPROVED by FSUE "VNIIMS", December 07, 2004.

REGISTERED by FSUE "VNIIMS" of the Federal Agency on Technical Regulation and Metrology, December 07, 2004.

INTRODUCED for the first time

CONTENTS

STATE SYSTEM FOR ENSURING UNIFORM MEASURMENT.

MI 2891 - 2004

GENERAL SOFTWARE REQUIREMENTS FOR MEASURING INSTRUMENTS.

1. SCOPE

1.1 The present Recommendation establishes general requirements for software of measuring instruments (MI) and is applied during software development, MI tests and identification (including recurrent identification).

1.2 The requirements of the present Recommendation are imposed on software of the following kinds:
- software which is part of a measuring system, functioning on the base of a personal computer;
- software which is an independent product that can be used for collecting, processing, storing and displaying of measuring information;
- software which is an integral part of MI (built-in software);
- software for controllers and computing units.

2. NORMATIVE REFERENCES

The present Recommendation uses principles and recommendations found in the following documents:

PR 50.2.009-94 STATE SYSTEM FOR ENSURING UNIFORM MEASUREMENT. The procedure of testing and type approving of measuring instruments.

GOST R 8.596-2002 STATE SYSTEM FOR ENSURING UNIFORM MEASUREMENT**.** Metrological assurance for measuring system. General Principles.

GOST R ISO/MEK 17025-2000. General requirements for the competence of testing and calibration laboratories.

GOST R ISO/MEK 12119-2000. Information technology. Software packages. Quality requirements and testing.

GOST R ISO 9127-94. Information processing systems. User documentation and information on the packing for commodity software packages.

GOST R 34.11-94 Information technology. Cryptographic protection of information. Hashing function.

GOST R 34.10-2001. Information technology. Cryptographic protection of information. Processes of generation and verification of electronic digital signatures

The European Union's Measuring Instruments Directive 2004/22/EC.

Recommendation OIML R76. Non-automatic Weighing Instruments.

Recommendation OIML D-SW 0.11 General requirements for software of measuring instruments.

WELMEC 7.1 Software Guide 7.1. Software requirements based on the Measuring Instruments Directive.

КООМЕТ Recommendation R/LM/10:2004 Software of measuring instruments. General technical requirements.

3. GENERAL PRINCIPLES

3.1 The present document establishes requirements for metrological control of MI software in the form of general requirements for software documentation, structure, identity to software certified during type approval tests, inaccuracy, and protection.

3.2 The Recommendation is intended for use in organizations that develop software, test MI for type approval, test (certify) software, or use software to collect, process, store and display measurement information.

3.3 Software testing (certification) is carried out by the State Testing Centers of Measuring Instruments (STC of MI), empowered to test MI for type approval by the federal executive authority body, which renders state services in the field of technical regulation and metrology and by bodies of MI software certification system. Specialists working in software developing organizations may be invited to participate in tests.

4. TERMS AND DEFINITIONS

Software of measuring instruments is a computer program or a totality of programs for collecting, transmitting, storing, and displaying of measurement information, as well as software documents necessary for these programs to function.

Data is measurement information presented in a form suitable for transmitting, interpreting or processing.

Software separation is isolation (description) of functions and parts subject to metrological control in MI software which falls into the field of the State Metrological Control and Supervision (SMCaS).

Interface is a common border between two blocks with different characteristics pertaining to functions, physical connections and information exchange.

User interface is the interface which provides information exchange between the user and the components of technical or software instruments of the information processing system.

Protected interface: an interface is considered protected:

- if a defined set of parameters, data and functions of a program can be sent or changed only through this interface;
- if it is impossible, by using this interface, to input into software commands or data, which are vaguely defined and may be mistakenly interpreted as measurement results, or commands, which may be used for the distortion of the displayed, processed and stored measurement results or other controlled data, or for the unauthorized adjustment or modification of software settings.

Inadmissible software modifications are changes to software (or to its parts) subject to metrological control;

Approved software is software of measuring instruments, which have passed type approval tests, and software certified during these tests.

Protected software and data - software and data, any modification of which is either impossible or becomes apparent during its start or operation.

Software identification is a checkup and verification of software integrity and authenticity**.**

Software and data integrity is a condition of software and data, when any intentional or accidental changes are absent.

Data authenticity is a condition of data, when their origin can be verified and when they can be unambiguously ascribed to certain measurements.

Checksum is the addition of all bytes of a program code or a data set. To obtain a result with a fixed number of figures, a modulo addition is often used. Checksum is often used as a simple hash code (GOST R 34.11-94).

Hash code is a result of an arithmetical operation on all bytes of the program code The result of a hashing algorithm includes only some bytes, and the algorithm is designed in such a way that any modifications to the program code or data will lead, with a high probability, to a different result, i.e. the changes will become apparent.

Electronic signature is an electronic signature of a file (program code or data) which is generated in two steps: firstly a hash code is calculated and secondly the hash code is encrypted (GOST R 34.10-2001). Electronic signature is normally added to the program code or data set it has been generated from.

Software inaccuracy is the difference of results obtained with software from results obtained with standard software, under the same conditions. Software inaccuracy may arise due to a poor choice of algorithms, their instability, the accumulation of rounding inaccuracy on the intermediate stages of calculations, the use of finite sums instead of infinite series during calculations, software errors and distortions during transmitting, processing and displaying of measuring information, etc. Software inaccuracy may be expressed in the number of lost digits of accuracy when compared to the standard results.

Standard software is software that meets the highest requirements for its accuracy and functional characteristics, verified (in some cases, using independent methods) during its multiple tests and use.

5. SOFTWARE SUBJECT TO THE METROLOGICAL CONTROL

5.1 General principles

5.1.1 In the structure of MI software which falls into the field of SMCaS, all program modules, subprograms, procedures, etc., which implement MI software functions subject to metrological control are described (isolated), i.e. the program gets separated into parts which are subject to the metrological control, and into those which are not. Such separation makes it possible to modify parts of software which are not subject to metrological control without violating software compliance with the approved type.

5.1.2 Developers can separate software on the program level, implementing it in the program structure.

5.1.3 Software separation is done by software developers falling into the field of SMCaS. It is recommended that this separation be done by all MI software developers. The correctness of separation is verified during software certification tests of MI for type approval.

5.2 Methods of separation

Two methods of software separation are used:

- *Low-level software separation* is a separation within the program, i.e. a separation on the level of the programming language (e.g., encapsulation in the object-oriented programming).
- *High-level software separation* is a separation on the operation system level. This means that only certain programs or libraries (e.g., DLL - Dynamic Link Libraries in Windows) perform functions subject to control.

5.3 Implementation of separation

5.3.1 In both low-level and high-level separation, all parts of the program (subprograms, procedures, functions, classes, variables, etc.), which are used to process measurement results or which affect them, or are used in auxiliary functions (such as data display, protection, storage and transmission, and software identification), are subject to metrological control

5.3.2 Other parts of software, variables or parameters (e.g., subprograms, libraries, procedures for the interaction with the operation system or peripheral PC devices (excluding MI) are not subject to control. Modification of these parts is allowed without notifying the STC of MI or software certification bodies.

Encapsulation is a technique that unifies both data and code that operates the data in one programming language element (object).

6. GENERAL SOFTWARE REQUIREMENTS FOR MEASURING INSTRUMENTS

General requirements are imposed on software types indicated in 1.2. They are listed in Table 1.

Table 1

№№ p/p	Requirement	Recommendation paragraph with explanations
1. Documentation		
1.1	Software submitted for tests (certification) should be accompanied with documentation in accordance with the requirements of this Recommendation.	6.1
2. Structure		
2.1	Software should be designed in such a way as to be insusceptible to inadmissible influence by other software.	6.2.1
2.2	Software subject to metrological control should be designed to be insusceptible to inadmissible influence via user (or other) interfaces.	6.2.2
3. Software compliance with the approved type (identification)		
3.1	After it has been approved (certified), software subject to metrological control should not be modified. In every MI, software identical to the approved one should be used.	6.3.1
3.2	To verify software compliance with the approved type and to verify its integrity and authenticity, software identification is used.	6.3.2
4. Software inaccuracy		
4.1	Software inaccuracy is evaluated during tests (certification). Its value should not exceed the limits established by the normative documentation and technical requirements.	6.4
5. Protection		
5.1	MI software should include means of detection, identification and/or elimination of errors (functional defects) and distortions, which falsify the integrity of results.	6.5.1
5.2	Software protection from accidental or unintentional modifications should be used. Changes to software and data should become apparent within a short time.	6.5.2

6.1 DOCUMENTATION

6.1.1 Software submitted for tests (certification) should be accompanied with documentation in accordance with the requirements of this Recommendation.

6.1.2 A list of documents accompanying software includes:

- a description of software design and functions, including data processing sequence;
- a description software functions and parameters, which are subject to metrological control;
- a description of computing algorithms used in the software and their program flowcharts;
- a description of software modules;
- a list of interfaces and commands for every interface, including a declaration of their completeness;
- a list, values, and functions of all commands, received from the keyboard, mouse and other input devices;
- a description of the implemented methods for software identification;
- a description of the implemented methods for software and data protection;
- a description of user interfaces, all menus and dialogs;
- a description of stored or transmitted data sets;
- a user's manual;
- required system and hardware specifications (if this information is not included in the user's manual);

6.1.3 A list of documents accompanying software can be adjusted by agreement between the executor and the client of software tests (certification).

6.1.4 Graphic and text information in documentation should be designed to be suitable for complete and unambiguous understanding.

6.2 STRUCTURE

6.2.1 Separation of software

6.2.1.1 Software subject to metrological control should be developed to be insusceptible to inadmissible influence of other software.

Note:

Other software here is software that works together with software subject to metrological control (bound software), for example, various modules within the same program complex or parts of software which are not subject to testing (subprograms, procedures, etc.). Influence issues of programs which are not included in the list of approved software are considered in accordance with the requirement of 6.5.2.

6.2.1.2 In the structure of software there should be isolated (described) parts performing functions subject to metrological control, and parts performing functions which are not subject to metrological control, i.e. software separation is performed, in accordance with p.5. Interaction between these parts of software is done via protected software.

6.2.1.3 Separation of software is done for both the program product and the data.

6.2.1.4 The access to variable values of the controlled parts of software should not be available from software parts not subject to control, i.e. protection on the level of the programming language should be implemented (encapsulation).

6.2.1.5 Measurement information (measurement results) is sent only to the parts of software subject to the metrological control. Measurement results may be sent to the parts of software which are not subject to control only after they have been finalized by the controlled software parts.

6.2.5.If parts not subject to control have been modified, software separation should be preserved.

6.2.2 Software interfaces

6.2.2.1 Software subject to metrological control should be designed to make it insusceptible to any inadmissible influence via software (or other) interfaces. Commands received via software (or other) interfaces should influence software functions and data only in the way described in the documentation.

6.2.2.2 Modifying or distorting measurement results by means of software interfaces is not allowed, i.e. software interfaces should be protected.

Note:
To implement a protected interface, a command interpreter may be used which filters commands and data input. Inadmissible commands and data should not affect measurement information.

6.2.2.3 Software documentation should describe the purpose and functions of all commands of software (or other) interfaces.

Note:
If the interface for inputting commands into software is absent, this rule is not applied.

6.3 SOFTWARE COMPLIANCE WITH THE APPROVED TYPE (IDENTIFICATION)

6.3.1 Software integrity and data authenticity

6.3.1.1 After the MI type has been approved, software of these MI subject to metrological control may not be changed. For every MI, software identical to the approved one is used.

6.3.1.2 Any violations of software separation, changes to the software interface and to parts subject to metrological control, are not allowed. Modifying the controlled parts of MI software results in violation of the MI type approval results and calls for the new testing of MI in question and its software.

Notes:
1. After the MI type approval, the developer may modify parts of software not subject to control, however, the requirements of 5.3.1 should be met.
2. If parts not subject to control have been modified, the software documentation should be accordingly updated.

6.3.1.3 For a MI that has passed type approval tests, only
the approved software is used.

Notes:
1. User interface of MI software falling into the field of SMCaS, reports on measurement results, files with data include information on the software approval and its version number;
2. Software operation is possible only after it has successfully passed its identification procedure (see p.6.3.2).

6.3.1.4 The violation of data separation and modification of data parts subject to metrological control are not permitted. Data integrity and authenticity is evaluated and verified during software tests (certification) by means of their identification.

6.3.2 Software identification

6.3.2.1 Software identification is used to evaluate software compliance with the approved type, and to evaluate and verify the integrity and authenticity of software and data

.

Note:

A technical implementation of the compliance verification method can be the software version number, which should consist of two parts in accordance with the software separation principle. The first part shows the condition of software parts subject to metrological control and is calculated accordingly as a checksum (CRC-16, CRC-32) or a hash code (SHA-1, MD5, etc.) based on the controlled parts of the software. The second part shows the condition of parts not subject to control. This part of the version number is set by the developer. Any modification to the legally controlled software parts automatically results in the change of the first part of the version number.

6.3.2.2 Identification algorithm is part of software subject to metrological control and is protected. Identification should take place when the software starts. There should also be a way to do it via the user's command.

Note:

During the identification, the operation system and low-level drivers (video-, printer drivers, etc.) are not taken into account; however, drivers that ensure software operation with the MI should be identified.

6.3.2.3 If any changes are made to software parts subject to the metrological control, the software needs a new approval and new identification.

6.3.2.4 If the separation is not present, the whole software is subject to identification.

6.4 SOFTWARE INACCURACY

6.4.1 Software inaccuracy is evaluated during tests (certification). Its value should not exceed the limits established by the normative documentation and technical requirements.

6.4.2. Software developed to function as part of or together with a MI is evaluated for inaccuracy. The values of inaccuracy are established individually for every type of MI that includes software.

6.4.3 The software inaccuracy is evaluation in accordance with the test methods (programs) agreed upon between the client and the executor of tests (certification).

Notes:

1. Universal software (e.g., electronic tables, mathematical and statistical programs) used for the processing of measurement results may be considered evaluated and compliant with the present requirement. In this case, no evaluation of inaccuracy is required, however, the configuration/modification of the implemented software should be verified.
2. Methods of software inaccuracy evaluation may include:
   developing standard (high-precision) software;
   developing (generating) standard data sets;
   comparing software with the approved software product (standard software);

evaluating inaccuracy when using universal software packages as standard software.

6.5 SOFTWARE AND DATA PROTECTION

6.5.1 Protection from errors

Software, subject to metrological control, should include means of detection, identification and/or elimination of errors (functional defects) and distortions, which can compromise the integrity of measurement results.

Notes:

1. In case of error, the user should be informed about it. The program displays an alert (visual and/or sound signal) requiring to terminate work until the error is eliminated.
2. If it is impossible to identify and/or eliminate the error, the program should be able to terminate abnormally and log this event at the following start. Program results, interfered by the error, should not be used from that point on.

6.5.2 Software and data protection from modifications

6.5.2.1 Software and data subject to metrological control should be protected from accidental or unintentional modifications. Changes to software and data should become apparent within a short time.

6.5.2.2 To meet this requirement, software and data can be protected from modifications using simple means, for example, a text editor.

Notes:

1. The reasons for accidental or unintentional changes are for mostly due to service errors of software and/or MI.
2. Intentional modifications of software (and/or MI) and data, using special software means, are an infringement of the law.

6.5.2.3 Data (files) containing measurement results should be protected from any distortions.

Note:
Data should be protected by means of digital signatures or encryption.

6.5.2.4 Software subject to metrological control should be protected
from any modifications and influence by persons who have no clearance to work with this software (unauthorized persons). Before starting to work with software, a user should receive appropriate rights for work (should be authorized).

Note:
At the program start, a user is requested to enter his or her account details (last name and password). After three unsuccessful attempts to log into the account , the program terminates and an entry is written in a special log file which identifies the event(unsuccessful attempt to authorize), its date and time.

6.5.2.4 Program code and parameters subject to metrological control should be protected.

Notes:

1. Protection of program code subject to metrological control is defined in accordance with requirements of 6.5.2.1 and 6.5.2.2.
2. Protection of software parameters is carried out in accordance with 6.5.2.3, as well as by means of implementing a software counter or event log, containing:

information about any changes in software parameters (date and time, user information);
all previous configurations of software parameters;

3. The event log should be protected in accordance with p. 6.5.2.2.

7. LEVELS OF REQUIREMENTS

The following levels of requirements are assigned to the software of MI according to each kind of requirements: low, medium, and high.

The levels are assigned by SCT of MI or by a body which carries out software certification, as agreed with the client of software tests (certification).

When assigning the levels or requirements, technical features of MI and their purpose are taken into account, leading to the assignment of software requirements in various degrees.

MI software is evaluated according to the chosen levels of:

- intensity *of examination,*
- degree *of* conformity *(identification),*
- strength *of protection.*

7.1 Intensity of software examination

Low	Software functions are examined during standard MI type approval tests according to the tests program.
Medium	Software is tested on the basis of the program functions description submitted by its manufacturer. Software influence on measurement results and its means of identification and protection are evaluated.
High	In addition to regular tests, which indentify metrological characteristics and the correct operation of performed functions, the software source code is tested. The subject of the program source code test can be, for example, the implementation of computing algorithms.

7.2 Degree of software conformity (identification)

Low	The implemented software of an individual MI should be in conformity with the approved one.
Medium	In addition to the low conformity level, in some cases depending on the technical features, certain software parts may be defined as "unalterable" at type approval. Unalterable parts shall be identical to the approved software in every MI.
High	Every MI should use software completely identical to the approved one.

7.3 Software protection

Low	No special protection of the controlled software and data from inadmissible modifications is required.
Medium	Software and data subject to metrological control are protected from inadmissible modifications using simple program means, like for example, text editors.
High	Software and data subject to metrological control are protected from inadmissible changes using special program means (debuggers and hard disc editors, software development tools, etc.).

**FEDERAL STATE UNITARY ENTERPRISE
THE ALL-RUSSIAN RESEARCH INSTITUTE
OF THE METROLOGICAL SERVICE (FSUE "VNIIMS")**

APPROVED by

Director of Institute

______________S.A. Kononogov

«   »   2004 г.

RECOMMENDATION

STATE SYSTEM FOR ENSURING UNIFORM MEASURMENT

GENERAL SOFTWARE REQUIREMENTS FOR MEASURING INSTRUMENTS

MI 2891 - 2004

**Moscow
2004**