XIX IMEKO World Congress
Fundamental and Applied Metrology
September 6−11, 2009, Lisbon, Portugal

AN INTERNATIONAL HARMONISED MEASUREMENT SOFTWARE GUIDE: THE NEED AND THE CONCEPT

Norbert Greif [1], Graeme Parkin [2]

[1] Physikalisch-Technische Bundesanstalt (PTB), Abbestraße 2-12, 10587 Berlin, Germany, Norbert.Greif@ptb.de
[2] National Physical Laboratory (NPL), Hampton Road, Teddington, TW11 0LW, United Kingdom, Graeme.Parkin@npl.co.uk

Abstract − Software is an intrinsic part of measurement. It is used in instruments to control experiments, store and process measurement data, analyse and display results and to implement mathematical techniques. Some innovations in measurement have been enabled through the use of software for simulations or complex analysis. For example, the international temperature scale ITS90 requires the processing of high order polynomials and can only be implemented using software. Given this reliance, improvements in the quality of software and reduced cost of its development are vital to the effective delivery of metrology.

Keywords: measurement software, software quality, software development, software assessment

1. INTRODUCTION

The Physikalisch-Technische Bundesanstalt (PTB) and the National Physical Laboratory (NPL) have been working on how best to produce or show that software is fit-for-purpose [1-3]. Over the last year they have been working together to develop a new international guide for the development and assessment of measurement software.

This paper gives the purpose and rationale for such an international guide. It also describes what work has been done and how we currently intend to make it international.

2. PURPOSE OF THE GUIDE

The purpose of the international guide is to enable

- developers of measurement software to know what they have to do to produce fit-for-purpose software, and
- assessors of measurement software to confirm that the developed software is fit-for-purpose.

By fit-for-purpose software we mean software that meets domain-specific measurement standards, relevant software quality standards and best software engineering practice.

The guide will also

- include a glossary of software terms to provide a common understanding of terminology in software development, software verification and validation and further essential phases of the software lifecycle;
- give descriptions of appropriate techniques to be used in the development and assessment of software;
- provide risk categories with the appropriate techniques to be used for each risk level (see section 4);
- provide checklists for developers and assessors; and
- where possible provide examples.

Although assessors of measurement software have a different perspective than the developers of measurement software, both are considered together. The reason for this is that the assessor needs to understand what the developer can reasonably provide to demonstrate the integrity of the measurement software. Equally, the developer needs to be aware of the legitimate concerns of the assessors in order to provide assurance in a manner that can be accepted by the assessors. In a competitive market, a consistent and transparent approach to quality assurance for measurement software is required.

3. THE NEED AND MAIN AIMS OF THE GUIDE

The following sections discuss the need and the main aims of the guide and the approach to deliver these aims. The common view of PTB and NPL is presented.

3.1. The need

Software is an intrinsic part of metrology. It is used in instruments to control experiments, store and process measurement data, analyse and display results and to implement many mathematical techniques.

Some innovations in metrology have been enabled through the use of software for simulations or complex analysis. For example, the international temperature scale ITS90 requires the processing of high order polynomials and can only be implemented using software. Given this reliance, improvements in the quality of software and reduced cost of its development are vital to the effective delivery of metrology.

However, due to the increasing complexity and dependency on software, there are considerable concerns over its quality. A study by NIST [4] stated that "Software bugs, or errors, are so prevalent and so detrimental that they cost the U.S. economy an estimated $59.5 billion annually". There is every reason to believe that Europe suffers in a similar way. NPL's recent audits of some instrument manufacturers, based on Software Support for Metrology (SSfM) Best Practice Guide 1, Validation of Software in Measurement Systems [3], and several examinations of measurement software carried out by the PTB's Software Testing Laboratory [2], have indicated that software engineering techniques are not widely used.

Today, there does not exist a comprehensive international software guide which can be used by measurement scientists and practitioners to overcome the deficiencies in software quality. A software guide that has been developed and accepted by leading NMI's would be more widely used and effective in the measurement community.

3.2. Main aims of the guide

In the following, the aims of the guide are summarised.

Who is the guide for?

The guide is aimed at those who implement and assess measurement software. These include at least measurement scientists, instrument manufacturers, testing and calibration laboratories.

Structure and type of the guide

Due to the complexity of software development it is expected that there will be one main guide and some supplementary guides. The supplementary guides will include more detailed information on specific software aspects, e.g. programming language style and coding standards or static analysis.

The main guide will be developed first and will be practical, short, and self-contained.

Types of measurement software

The guide will cover all types of measurement software including COTS (Commercial Off The Shelf) software, embedded software, control of instruments, mathematical processing and graphical user interfaces. Measurement software can be implemented in a laboratory or in an instrument.

Risk-based approach

No software can be shown to be completely error free due to the infeasibility of complete testing and impracticality of mathematical proof.

The application of various techniques can reduce the number of errors in the software, but the more techniques that are applied the more expensive the software is to develop. It is clear that software to be used in a safety-critical environment will require more effort than that in a non safety-critical one.

A risk-based approach provides a means to determine how much effort should be used in the development of software that is suitable for the type of software and for the consequences of when it goes wrong (see section 4).

Process view versus product view

By process view we mean gathering evidence during the development of the software to show that the software is fit-for-purpose as compared to testing the final (or some intermediate) software product as a black box.

The guide will consider both aspects of software quality and will concentrate on the process of providing evidence that the software product is fit-for-purpose.

Software lifecycle

The whole software lifecycle will be considered as it all affects whether the software is fit-for-purpose. To serve as the base for the software guide, a software process reference model is being derived from the international standard ISO/IEC 12207 [5].

Due to the aim of practicability, only the essential key process areas are being selected. Currently, it is proposed that the process reference model includes requirements analysis, design, implementation, testing, and operation/maintenance.

Structuring the software development process helps to categorise the diversity or the recommended development and assessment techniques, and the different activities of the lifecycle processes.

Relationship with other standards

There are many software standards covering different aspects of software. However, they do not cover what this guide will include. Where necessary, relevant software standards will be taken in consideration. Currently, these include: ISO/IEC 12207 [5], IEC 61508 [6], ISO/IEC 15504 [7], ISO/IEC 27005 [8], and the ISO/IEC 25000 series [9].

Wherever applicable, procedures, requirements, and recommendations of the guide will be traced back to relevant software standards.

4. CURRENT STATE OF THE RISK-BASED APPROACH

The guide will provide a risk assessment procedure based on the international standard ISO/IEC 27005 [8]. According to this standard, at first the guide will define risk categories based on measurement software specific risk factors with appropriate risk levels.

To keep simple the risk assessment and the following procedure of determining recommended measures (techniques) for each risk category, in a second step, the set of risk factors with different risk levels is mapped to a unified risk index, the so-called Measurement Software Index (MSI).

For each of the considered software lifecycle processes (see section 3), selected development and assessment techniques have to be assigned to the defined sublevels of the MSI.

To provide a simple risk assessment procedure, PTB and NPL are proposing the following restricted fundamental process:

- Initially, the basic risk factors are restricted to three:
    - level of ***control complexity*** (complexity of software interaction),
    - level of ***processing complexity***, and
    - level of ***system integrity***, which is considered to be composed of at least the criticality elements safety issues, security issues, or environmental issues.
    - However, the proposed risk factors and corresponding levels can be expanded by further domain-specific aspects if needed.
- The number of risk levels for each basic risk factor is restricted to four (very low, low, high, very high).
- The number of risk levels for the general Measurement Software Index MSI is restricted to five (0, 1, 2, 3, 4).

In the first part of the risk assessment procedure, for each of the three basic risk factors and for each of its four risk levels (very low, low, high, very high), a set of measurement software oriented characteristics has been drafted to derive the relevant risk level. Characteristics for the risk factor ***control complexity*** are, for example,

- the impact of software control functions on the measurement process,
- the influence of the software on the measurement result or
- the number and complexity of software interactions with other software/hardware subsystems.

Based on the risk-oriented characteristics of the three basic risk factors, a proposal for a general MSI level (a function of the basic risk categories) has been elaborated. Each combination of the basic risk factors, including the combinations of the different risk levels, has been mapped to a MSI of 0 to 4 (five levels).

Finally, these proposals for the applicable MSI levels have been summarised in a table to support the user in the risk treatment process. The proposed MSI levels need to be checked.

The remaining problem is the assignment of appropriate software development and assessment techniques to be used to the MSI levels. For each of the five MSI levels, the guide has to recommend which techniques and, specifically, what level of activity for each lifecycle process should be used. For that purpose, the following are being developed:

- a list of practical development and assessment techniques;
- appropriate levels of activities for each selected software lifecycle process (process requirements);
- assignments of selected techniques and process requirements to the MSI levels for each of the selected lifecycle processes.

Based on the agreed assumptions and the final decisions to be derived, a simple risk assessment procedure according to ISO/IEC 27005 [8] will be implemented by the guide. Thus, the user of the guide can ensure that the user's software is fit-for-purpose concerning the main risks of the user's specific domain.

5. WAY FORWARD: JCGM AD HOC GROUP ON MEASUREMNT SOFTWARE

An ad hoc working group has been set up under the Joint Committee for Guides in Metrology (JCGM) [10] which is associated to BIPM (The International Bureau of Weights and Measures). The main objective of the JCGM ad hoc group on measurement software [11] is the elaboration of a guide for the development and assessment of measurement software. The guide would assist

- developers of measurement software in the production of software that is fit-for-purpose; and
- assessors of measurement software in confirming that developed software meets its specification.

All drafts of the guide will be made available for review by as many interested persons and organisations as possible including at least the

- JCGM ad hoc group on measurement software;
- JCGM member organisations not represented in the JCGM ad hoc group on measurement software;
- National Measurement Institutes; and
- instrument manufacturers.

Further information regarding the JCGM ad hoc group are available from the open BIPM website [10, 11].

6. CONCLUSIONS

There is the need for an international software guide for metrologists, measurement scientists and practitioners which contains all that is required to develop fit-for-purpose software as currently none exists.

The concept of the international software guide is being jointly developed by PTB and NPL. It takes a risk-based approach. Further details of the concept are still being developed.

Currently, an ad hoc group on measurement software has been set up under the JCGM (Joint Committee for Guides in Metrology) to develop an international software guide for the measurement community.

REFERENCES

[1] N. Greif, G. Parkin, "A new guide for the development and assessment of measurement Software", *Advanced Mathematical and Computational Tools in Metrology and Testing (VIII)*, Paris, France, 23-25, June 2008.

[2] N. Greif, "Software testing and preventive quality assurance for metrology", *Computer, Standards & Interfaces*, 28(2006)3, pp. 286-296, Elsevier, March 2006.

[3] R. M. Barker, B. Wichmann, G. Parkin, *SSfM Best Practice Guide No 1*, "Validation of software in measurement systems", DEM-ES 014, National Physical Laboratory, January 2007.

[4] RTI, "The economic impacts of inadequate infrastructure for software testing", National Institute of Standards and Technology, 2002.

[5] ISO, ISO/IEC 12207:2008, Systems and Software Engineering - Software Lifecycle Processes, 2008.

[6] IEC, IEC 61508:1998, Functional safety of safety-related systems, parts 1-7, 1998.

[7] ISO, ISO/IEC 15504-4:2004, Information Technology - Process Assessment, part 4: Guidance on use for process improvement and process capability determination, 2004.

[8] ISO, ISO/IEC 27005:2008, Information Technology - Security Techniques - Information security risk management, 2008.

[9] ISO, ISO/IEC 25000:2005, Software Engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Guide to SQuaRE, 2005.

[10] Joint Committee for Guides in Metrology (JCGM), http://www.bipm.org/en/committees/jc/jcgm/.

[11] JCGM ad hoc group on measurement software, http://www.bipm.org/en/committees/jc/jcgm/adhoc_ms.html.