GROUP ON
EARTH OBSERVATIONS

QA4EO

A QUALITY ASSURANCE
FRAMEWORK FOR
EARTH OBSERVATION

A guide to establishing validated models, algorithms and software to underpin the Quality Assurance requirements of GEO

QA4EO-QAEO-GEN-DQK-005

QA4EO guide: QA4EO-QAEO-GEN-DQK-005 Community Approved

A guide to establishing validated models, algorithms and software to underpin the Quality Assurance requirements of GEO

Author:	Nigel Fox
E-mail:	Nigel.Fox@npl.co.uk
Editor:	Marie-Claire Greening
E-mail:	marie-claire@greeningconsulting.co.uk

Issued under Authority of:	QA4EO
Issue no:	Version 4.0
Date of issue:	27 March 2010

Summary of Changes since last issue:

Version 3.0 now includes edits and comments made on version 2.0 by GSICS in March 2009.

Version 4.0 incorporates edits and comments made on version 3.0 by WMO and GSICS August – November 2009.

Table of Contents

1 Abstract

This document provides the link to existing generic "best practise" guidelines for testing and validating models or algorithms and derived software used for measurement systems. These guidelines are recommended for use within the processing chain of Earth Observation (EO) data products. Many guidance documents exist to aid in the writing and development of software. Some have the status of international standards and others are in-house procedures, but all in concept are broadly similar to each other. However, few of these provide detailed guidance on appropriate techniques for validation and assessment of its "fitness for purpose" and the means to assign a "quality indicator". Those linked from this QA4EO key guideline document provide this guidance for both the developed software and are the models or algorithms upon which it is based.

2 Scope

Models, and the subsequent algorithms or software, are crucial to the production of practical EO data products with their role increasing in importance as the data product progresses along the chain from Level 1 to Level n. It is well understood that the complexity and lack of understanding of many bio-geo-physical processes makes this topic one of the most challenging within the Earth Observation (EO) sector. Consequently the establishment and assignment of appropriate quality indicators is often a non-trivial exercise. However, if users are to make full use of knowledge information products, efforts must be made to establish representative and realistic quality indicators for all the processing steps to complement those on the source data itself. To achieve this may sometimes require "best guess" estimates on the uncertainties and, as a consequence, the resultant product may have large error bars. Developers and users must recognise this and steps must be taken to educate the community to interpret this correctly. In carrying out this process the most important aspect is to document the decision process and reasoning behind the decision.

The "best practice" guidelines linked from this document concern the development of validated software, models and algorithms used within the data processing chain. Often such software already exists and much of what is proposed here may not easily be applicable retrospectively. The focus of the guides referenced by this document concern the approach and methods associated with validation. General guidance on the development of software is available from many sources (e.g., the European Community space standard ECSS-E-40B [1]) and, whilst there may be merit in its harmonisation, that is not the principle aim of the QA4EO activity.

The guidelines provide a range of different validation techniques together with a "risk-based" method to identify the most appropriate method for particular applications. To aid the user, checklists are provided to facilitate development and also to aid any subsequent audit. Given the complexity of this topic, this document recommends a set of complementary guidelines. Some overlap in their scope, but together they form a comprehensive Quality Assurance (QA) framework in their own right.

3 Terminology

All terms within this document are based on internationally-agreed definitions that are, in many cases derived directly from formal standardising bodies such as the International Organization for Standardization (ISO). These agreed definitions can be found on the QA4EO website (http://QA4EO.org).

4 Background and Context

This key guideline is written as part of a set, based on the adoption of existing best practise, to form a Quality Assurance Framework for Earth Observation (QA4EO). The QA4EO was developed to meet the current and aspirational needs of the societal themes of the Group on Earth Observation (GEO)'s Global Earth Observation System of Systems (GEOSS). It was prepared as a direct response to GEO task DA-06-02 (now DA-09-01-a) to "Develop a GEO data quality assurance strategy, beginning with space-based observations and evaluating expansion to *in situ* observations, taking account of existing work in this arena".

This document is written to provide an interface and brief introduction to a set of generic "best practise guides" that together form a linked subset of a broader set of key guidelines that form the Quality Assurance Framework for Earth Observation (QA4EO).

5 Outcomes

The document provides a link to a set of "best practise" guidelines that can be used during the development of algorithms, models and software to ensure it is appropriately validated.

6 Inputs

There are no specific identifiable inputs required by this procedure to enable the outcomes to be achieved.

7 Standards and Traceability

The procedure outlined in this document has no quantitatively assessable outcomes in its own right and is simply a link to other guidelines. However this document may make reference to the use of other documentary standards. As time progresses new documentary standards and "best practises" may be developed and adopted by QA4EO that are applicable to the activities described in this document. Therefore, it is recommended that the latest version of this document be reviewed for changes prior to its use.

1. QA4EO endorsed vocabulary http://QA4EO.org

2. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=45315.

3. BIPM MRA http://www.bipm.org/

Any comparison carried out following these guidelines will select and make use of appropriate "reference standards" to perform the task that they describe.

8 Guidelines for validating models, algorithms and software

8.1 Introduction

The top-level approach to a framework for QA / Quality Control (QC) methods and procedures for software development or, more specifically "data processing", covers the different stages of building a software system: model, algorithm and implementation via software. The framework describes the steps to be followed so that the software system can be validated. This would include assessment, auditing and certification of the validation process if required. The framework focuses on the recommended validation techniques, i.e., techniques to be applied during the software lifecycle to improve the quality of the system and to provide evidence that can be used to assess the validation process.

The overall approach of the framework follows the approach to validation in Best Practice Guide No. 1 "Validation of Software in Measurement Systems [2]". This guide uses risk analysis to determine a level for the software. The software level is used to determine what techniques need to be applied in the life cycle of the software in order to achieve the necessary quality of software required by the risk analysis, i.e., its "fitness for purpose"). The measurement software level from the guide can be adapted as a quality indicator for the project.

The approach to validation follows a risk-based approach. This means that the effort spent to assure the correct operation of the system increases with the complexity of the system and with the impact of the system operating incorrectly. This approach is implicit in all practical validation methodologies and is explicit in industry-specific schemes such as DO 178B for civil aircraft industry software or the safety critical system standard IEC 61508. The development of the Best-Practice Guide No. 1 [2] was based on previous risk-based validation approaches; the guide is aimed at measurement systems, but is more generally applicable.

The validation processes and application of recommended techniques should not be "bolted-on" at the end of the software development and testing. Most of the recommended techniques must be employed during design, development and testing. It would be ineffective or not at all cost effective if applied afterwards. The only step that should appear outside the process is to assess that the validation techniques achieved their aims. This final step can be achieved through auditing.

8.2 Software QA guidelines

8.2.1 Introduction

In general, the processing of data consists of a set of linked stages where each stage can be considered as a separate system. Each system needs to be considered separately, as does the models and algorithms implemented by the software within each system. Each software system typically has inputs consisting of data (processed by earlier systems) and quality indicators for that data (assigned by earlier systems). Similarly, the outputs from each software system are the processed data and new quality indicators for the new data. These new quality indicators will depend on the input data, the input quality indicators and processing that has been done.

The development and validation of each software system dealing with (structured) scientific data has three stages:

1. Development and validation of a model to adequately describe the input and output data,

2. Development and validation of algorithms to "solve" the model for the expected input data, and

3. Development and validation of software to implement the model and algorithms.

At each stage, the model must describe how the output quality matrices depend on the other data and the algorithms must describe how to calculate the output quality indicator values. In practice, the different stages will overlap in the overall process. Each "development and validation" stage consists of a number of steps:

- Risk analysis to determine the choice of appropriate validation techniques.
- Development, based on chosen validation techniques.
- Assessment that the validation techniques have delivered the desired "level" of system.

8.2.2 Model validation

In modelling, the functional model that describes the functional relationships between quantities (including those that are measured) must be distinguished from the statistical models that describe the probability distributions (and the uncertainties) associated with those quantities. The particular issues of "continuous" models must also be considered where the solutions consist (mathematically) of a continuum of values and solved by finite element methods and similar techniques.

The recommended guides consider various aspects of modelling and the means to select appropriate validation techniques.

- Correctness of the functional model: does the model capture the underlying science?
- Comprehensiveness of the model: if a model is too simple it may not capture the essentials of the underlying science, although too detailed a model may be impossible to solve.
- Statistical model: does the model correctly describe the uncertainties and correlations associated with the quantities in the functional model?

In many cases, a well-established model will be appropriate and the model validation will simply be by citing established references.

The guidelines that should be followed are Best Practice Guide No. 4 “Discrete modelling and experimental data analysis”[3], which follows Best Practice Guide No. 1 in its risk-based approach to validation [2], and NPL Report CMSC 29/03 “Model Validation in Continuous Modelling” [4], which deals with the particular issue of continuous models.

8.2.3 Algorithm Validation

An algorithm is a mathematical specification of the computational steps to perform a calculation, e.g., to determine the output quantities given a physical model. Although an algorithm is expressed in mathematics, it is understood that it operates using finite precision arithmetic on finite precision data. Therefore, different algorithms whose specifications are mathematically equivalent may behave in different ways numerically. The validation of algorithms is intimately linked with the testing and validation of software implementations of the algorithms, as described in Good Practice Guide No. 16 “Testing and Validation of Algorithms and Software” [5].

In many cases, a well-established and documented algorithm will be appropriate and the algorithm validation will simply be by citing established numerical analysis texts.

8.2.4 Software Validation

Software development should follow the procedures of the European Community space standard ECSS-E-40B[1] or similar. However, this standard does not specify particular techniques to assure that software is "fit for purpose" and the necessary evidence to enable a QI to be robustly assigned to its outputs. For this, following the methodology of Best Practice Guide No. 1 [2] is recommended.

8.2.5 Assessment, and Auditing

8.2.5.1 Evidence

All the validation techniques that are used at all stages of the software system must be documented and their results recorded. This provides evidence of the use of the validation technique and can be used to demonstrate that the software system has been properly developed. Also, the risk analysis and the reasons for the choice of validation techniques must be recorded. A pro-forma (electronic or paper) can easily be produced for the collation of such evidence.

8.2.5.2 Assessment

To assess that sufficient and appropriate validation techniques have been applied, the choice of techniques and the evidence from those techniques must be examined. This assessment should be based on a sound technical understanding of modelling, numerical analysis and software engineering (respectively). This should initially be carried out in-house by an "expert" independent of the development. Subsequently, this can be checked by external audit if required by the customer, either through sampling or in-detail, depending on the criticality of the output.

8.2.5.3 Auditing

Ideally the functionality and performance of software systems should require third-party assessment of the validation steps carried out. This can easily be performed by auditing the documentation of the risk analysis and the evidence from the application of the validation techniques; as opposed to the need for any direct observation or repetition of the validation techniques. Best Practise Guide No. 1 [2] includes audit checklists that can be applied to the validation of software based on techniques in that guide.

8.3 References

1. European Community space standard ECSS-E-40B http://www.ecss.nl/

2. Best Practice Guide No. 1 "Validation of Software in Measurement Systems http://www.npl.co.uk/server.php?show=nav.1210#2

3. Best Practice Guide No 4 "Discrete modelling and experimental data analysis" http://www.npl.co.uk/server.php?show=nav.1210#2

4. NPL Report CMSC 29/03 "Model Validation in Continuous Modelling" http://www.npl.co.uk/server.php?show=nav.1210#2

5. Good Practice Guide No 16 "Testing and Validation of Algorithms and Software". http://www.npl.co.uk/server.php?show=nav.1210#2

9 Conclusion

This document has introduced the basic principles and described in detail within the linked guidelines the processes that can be followed to establish validated software, algorithms and models. It is anticipated that "best practise" may already be in common use within the community, or at least a good portion of it. However, harmonisation of the processes and evaluation criteria would clearly make it easier for project teams to review progress and adequacy of any software or model system and the products derived using them. Transparency of the processes with common terminology and techniques would also make it simpler to merge software products from different developers and also more easily allow upgradeability and isolation of operational anomalies.