Living up to standards

Margaret King
TIM/ISSCO
ETI
University of Geneva
Margaret.King@issco.unige.ch

Abstract

This paper attacks one part of the question "Are evaluation methods, metrics and resources reusable" by arguing that a set of ISO standards developed for the evaluation of software in general are as applicable to natural language processing software as to any other. Main features of the ISO proposals are presented, and a number of applications where they have been applied are mentioned, although not discussed in any detail.

Acknowledgements

The work recorded here is far from being all my own. I would like first to record my thanks to Nigel Bevan, technical editor of the ISO standards discussed for much interesting and enlightening discussion. Then many thanks must go to all my colleagues in the EAGLES and ISLE projects, especially Sandra Manzi and Andrei Popescu-Belis. Finally, I must thank all those whose work on applying the standards reported here provoked reflection and helped to convince me of the value of the approach: Marc Blasband, Maria Canelli, Dominique Estival, Daniele Grasso, Véronique Sauron, Marianne Starlander and Nancy Underwood.

1 Introduction

This paper is constructed around a syllogism:

1. ISO standards 9126 and 14598 are applicable to the evaluation of any type of software
2. Natural language processing software is a type of software
3. ISO standards 9126 and 14598 are applicable to the evaluation of natural language processing software.

In support of the major premise, I shall set out some of the major features of the ISO standards in question. The minor premise needs no support: indeed, it is almost a tautology. The truth of the conclusion will logically depend therefore on whether I have managed to convince the reader of the truth of the major premise. There will be little explicit argument in this direction: simply setting out key features of the approach should suffice. I will try, however, to reinforce the conclusion by briefly reviewing a number of natural language processing applications where the ISO standards have been followed with encouraging results. My hope, of course, is to encourage readers to apply the standards themselves.

2 ISO standards work on software evaluation

ISO has been publishing standards on software evaluation since 1991. The bibliography gives a detailed picture of what standards have already been published and of what standards are in preparation. ISO/IEC 9126 was the first standard to appear. It has subsequently been modified, and

in its new versions the original content of 1991 has been refined, modified and distributed over a series of separate but inter-related standards.

The keystone of ISO work is that the basis of an evaluation is an explicit and detailed statement of what is required of the object to be evaluated. This statement is formulated very early in the process of defining an evaluation and is called a "quality model". The process of evaluation involves defining how measurements can be applied to the object to be evaluated in order to discover how closely it meets the requirements set out in the quality model.

"The object to be evaluated" is a clumsy phrase. It has been used because, in the ISO picture, evaluation may take place at any point in the lifecycle of a software product, and may have as its object not only the final product but intermediate products, including specifications and code which has not yet been executed. It follows from this that a quality model may apply to a set of specifications just as much as to a piece of finished software. Indeed, one might envisage using quality models as a way of guiding the whole process of producing a software product, from initial research and prototyping through to delivering and field testing the final product. That this is in line with best practice in software engineering constitutes, to my mind, an argument in favour of the ISO proposals.

As well as a set of standards relating to the definition of quality models (the 9126 series) ISO also offers a set of standards relating to the process of evaluation (the 14598 series). One document sets out a standard for the evaluation process seen at its most generic level, further proposals relate definition of the process to the particular viewpoints of software developers, of acquirers of software and of evaluators typically working as third party evaluators. Other documents in the 14598 series provide supporting material for those involved in evaluation, offering standards for planning and management of evaluations and for documentation of evaluation modules. Of the 9126 series, only the first document which directly deals with quality models has as yet been published. Documents in preparation deal with standards for the metrics which form a critical accompaniment to any quality model. It would be unrealistic in the space of a single paper to discuss even the documents already published in any detail. In what follows, we concentrate on outlining the foundations of the ISO proposals, the quality model and the process of evaluation.

3 Quality models (ISO 9126)

A quality model consists of a set of quality characteristics, each of which is decomposed into a set of quality sub-characteristics. Metrics measure how an object to be evaluated performs with respect to the quality characteristics and sub-characteristics. The quality characteristics and sub-characteristics making up the quality model of ISO 9126-1/01 are shown in figure 1, on the next page. All that figure 1 shows are names: ISO 9126-1/01 gives both definitions and discussion.

The quality characteristics are intended to be applicable to any piece of software product or intermediate product. They are thus necessarily defined at a rather high level of generality, and need to be made more specific before they are applicable to any particular piece of software. They are also defined through natural language definitions, and are thus not formal in the mathematical or logical sense. This being so, they are open to interpretation. Defining a specific evaluation implies deciding on an appropriate interpretation for that evaluation.

ISO 9126/01, whilst not barring the possibility that a quality model other than that contained in the standard might be used, requires that if another model is used, it should be clearly described.

"*Software quality shall be evaluated using a defined quality model. A quality model shall be used when setting quality goals for software products and intermediate products. This part of ISO/IEC 9126 provides a recommended quality model which can be used as a checklist of issues relating to quality (although other ways of categorising quality may be more appropriate in particular circumstances). When a quality model other than that in this part of ISO/IEC 9126 is used it shall be clearly described*." (ISO 9126/01, 1.5, Quality relationships).

Work within the EAGLES project on defining a general framework for evaluation

design extended this model by allowing the quality sub-characteristics in their turn to be decomposed; the process of decomposition being repeated if necessary.

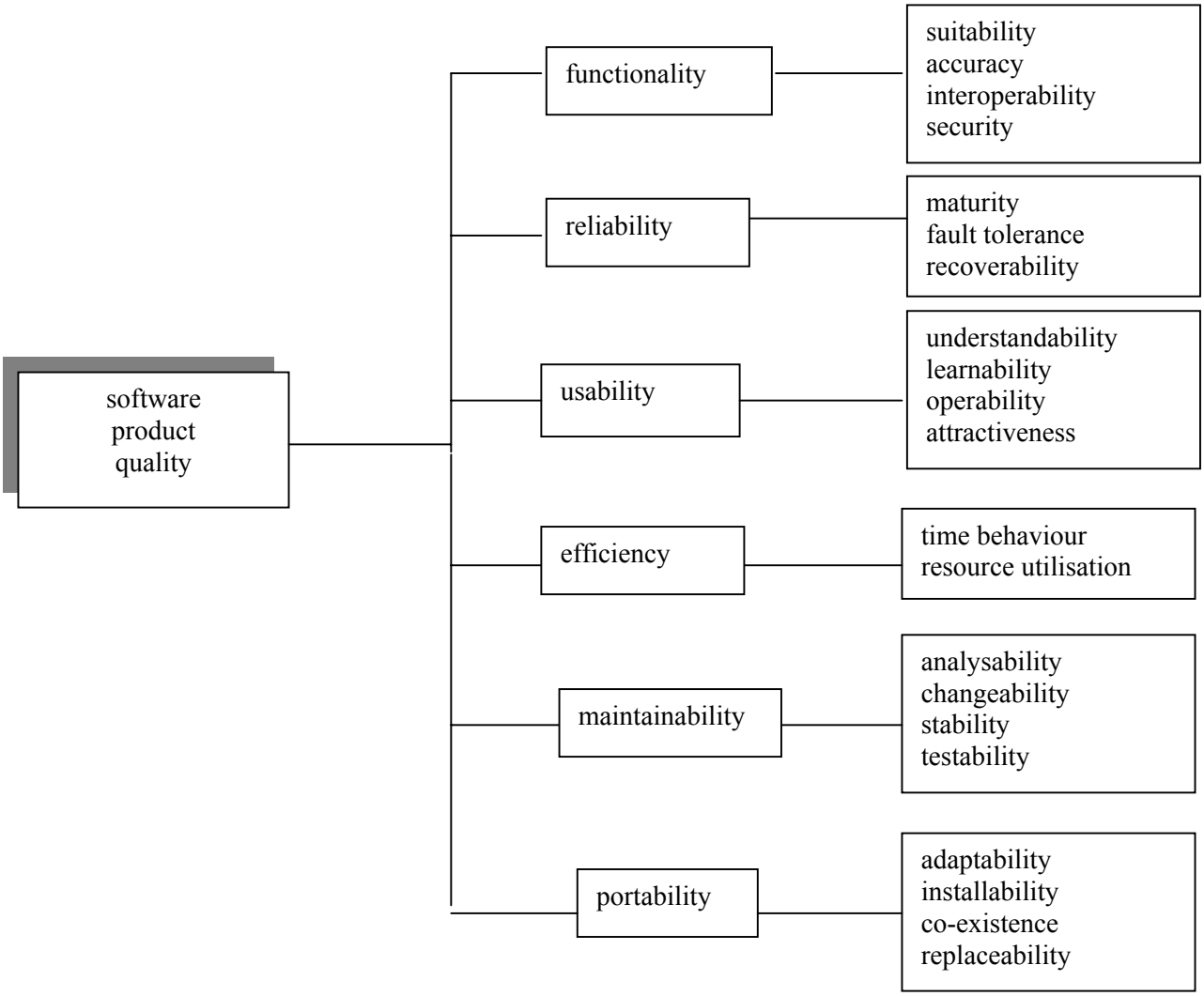

Figure 1

The structure thus obtained is hierarchical, and, theoretically of unlimited depth. ISO 9126-1/01 does not rigidly specify the relationship between quality characteristics and metrics. The EAGLES extension requires that each terminal node of the structure has at least one metric associated with it. The structure then becomes a hierarchy of attribute value pairs, where each node is labelled with the name of an attribute. The values of the attributes at the terminal nodes are directly obtained by the application of metrics. The value of a higher level node is obtained by combining the values of attributes nodes immediately dominated by the higher level node: values percolate upwards. Exactly how the combination of values is done is determined by a combining function which reflects the relative importance of the attributes in a particular evaluation. This formalization provides an operational semantics for any particular instantiation of the quality model. Once the evaluation designer has decided what attributes to include in his quality model

and how to organise them, and once he has defined and assigned metrics to the terminal nodes, what functionality, for example, means within that quality model is defined by the decomposition of the functionality node and by the associated metrics.

Metrics will be discussed only briefly here. The ISO standard distinguishes between internal metrics, external metrics and quality in use metrics. The difference between them is determined by what kind of an evaluation object they are applied to.

Internal metrics apply to static properties of software, that is software considered independently of its execution. Examples might be the number of lines of code or the programming language used. As can be seen from the inclusion of the programming language in this list, metrics are not necessarily quantitative in their nature, although they should, of course, be as objective as possible. (This is one of the points we shall not go into further here.)

External metrics apply to software when it is being executed, to the behaviour of the system as seen from outside. Thus they may measure the accuracy of the results, the response time of the software, the learnability of the user interface and a host of other attributes that go to make up the quality of the software as a piece of software.

Quality in use metrics apply when the software is being used to accomplish a particular task in a particular environment. They are more concerned with the effects of using the software than with the software itself. Quality in use metrics are therefore very dependent on a particular environment and a particular task. Quality in use is itself a super-ordinate aspect of quality, for these same reasons. It is clearly influenced by the quality characteristics which make up the quality model, but is determined by the interaction of different quality characteristics in a particular task environment.

The ISO standards published so far say little about what makes a metric a good metric. Some work elsewhere (Popescu-Belis, 1999, Hovy et al, 2003) has made some suggestions.

First, metrics should be coherent, in the sense that they should respect the following criteria:

- A metric should reach its highest value for perfect quality (with respect to the attribute being measured), and, reciprocally, only reach its highest level when quality is perfect.
- A metric should reach its lowest level only for the worst possible quality (again, with respect to the attribute being tested)
- A metric should be monotonic: that is, if the quality of software A is higher than that of software B, then the score of A should be higher than the score of B.

We might compare two metrics (or more strictly two rating functions: see the section on process below) by saying that a metric m_1 is more severe than a metric m_2 if it yields lower scores than m_2 for every possible quality level. Conversely, one metric may be more lenient than another.

To these rather formal considerations, we might add:

- A metric must be clear and intuitive
- It must correlate well with human judgements under all conditions
- It must measure what it is supposed to measure
- It must be reliable, exhibiting as little variance as possible across evaluators or for equivalent inputs
- It must be cheap to prepare and to apply
- It should be automated if possible

4 Evaluation process (ISO 14598)

A first section of ISO 14598-1/99 is concerned with an overview of how all the different 9126 and 14596 documents concerned with software evaluation fit together. This overview can be summarized quite briefly. It is fundamental to the preparation of any evaluation that a quality model reflecting the user's requirements of the object to be evaluated be constructed. The 9126 series of documents is intended to support construction of the quality model.

The 14598 series is concerned with the process of evaluation, seen from different viewpoints. Separate documents in the series tackle evaluation from the point of view of developers, acquirers and (third party) evaluators. All of these make use of the 9126 series, and are further supported by the second half of 14598-1, which sets out a generic picture of the process of evaluation, and by two further documents, the first concerned with planning and management of a software evaluation process, the second with guidance for documenting evaluation modules.

Although these other documents in the series are clearly important, we limit ourselves here to summarizing the process of evaluation, as set out in ISO 14598-1.

The evaluation process is conceived as being generic: it applies to component evaluation as well as to system evaluation, and may be applied at any appropriate phase of the product life cycle.

The evaluation process is broken down into four main stages, each of which is considered separately below:

Stage I: Establish evaluation requirements.

This step is broken down into a further three steps:

a) Establish the purpose of the evaluation

The commentary on this point reveals just how wide the scope of the standard is intended to be. The purpose of evaluating the quality of an intermediate product may be to:

- Decide on the acceptance of an intermediate product from a sub-contractor
- Decide on the completion of a process and when to send products to the next process
- Predict or estimate end product quality
- Collect information on intermediate products in order to control and manage the process

(The reader will remember that intermediate product means, for example, specifications or code before it is executed).

The purpose of evaluating an end product may be to:

- Decide on the acceptance of the product
- Decide when to release the product
- Compare the product with competitive products
- Select a product from among alternative products
- Assess both positive and negative effects of a product when it is used
- Decide when to enhance or replace the product.

It follows from this very broad range of possibilities that the standard is meant to apply not only to any kind of intermediate or final software product, but to any evaluation scenario, including comparative evaluation.

b) Identify types of products to be evaluated

Types of products here does not mean application software, but rather is concerned with the stage reached in the product's life cycle, which determines whether and what intermediate product or final product is to be evaluated.

c) Specify quality model

The quality model is, of course, to be defined using ISO 9126-1/01 as a guide. However, a note quoted again below adds:

"The actual characteristics and sub-characteristics which are relevant in any particular situation will depend on the purpose of the evaluation and should be identified by a quality requirements study. The ISO/IEC 9126-1 characteristics and sub-characteristics provide a useful checklist of issues related to quality, but other ways of categorising quality may be more appropriate in particular circumstances." (ISO 14598-1/99)

An important word here is "checklist": the basic purpose of the ISO quality model is to serve as a

guide and as a reminder for what should be included in evaluating software. Arguing about the exact interpretation of the quality characteristics is pointless. Their interpretation is given by the model in which they are incorporated.

Stage II:**Specify the evaluation**

This too breaks down into three steps:

a) **Select metrics**

b) **Establish rating levels for metrics**

c) **Establish criteria for assessment**

Quality characteristics and sub-characteristics cannot be directly measured. Metrics must therefore be defined which correlate to the quality characteristic. Different metrics may be used in different environments and at different stages of a product's development. Metrics have already been discussed to some extent in the section on quality models above.

A metric typically involves producing a score on some scale, reflecting the particular system's performance with respect to the quality characteristic in question. This score, uninterpreted, says nothing about whether the system performs satisfactorily. To illustrate this idea, consider the Geneva education system, where marks in examinations range from 1 to 6. How do you know, without being told, that 6 is the best mark and 1 the worst? In fact, most people guess that it is so: they may then have a difficult time in Zurich where 1 is the highest mark. Establishing rating levels for metrics involves determining the correspondence between the uninterpreted score and the degree of satisfaction of the requirements. Since quality refers to given needs, there can be no general rules for when a score is satisfactory. This must be determined for each specific evaluation.

Each measure contributes to the overall judgement of the product, but not necessarily in a uniform way. It may be, for example, that one requirement is critical, whilst another is desirable, but not strictly necessary. In this case, if a system performs badly with respect to the critical characteristic, it will be assessed negatively no matter what happens to all the other characteristics. If it performs badly with respect to the desirable but not necessary characteristic, it is its performance with respect to all the other characteristics which will determine whether the system is acceptable or not.

This consideration feeds directly into the third step, establishing criteria for assessment, which involves defining a procedure for summarizing the results of the evaluation of the different characteristics, using for example decision tables or weighting functions of different kinds.

Stage III: Design the evaluation

Designing the evaluation involves producing an evaluation plan, which describes the evaluation methods and the schedule of the evaluator action. The other documents in the 14598 series expand on this point, and the plan should be consistent with a measurement plan, as described and discussed in the document on planning and management. (ISO 14598-2/00)

Stage IV: Execute the evaluation

This final stage again breaks down into three stages:

a) **Measurement**

b) **Rating**

c) **Assessment**

These steps are intuitively straightforward in the light of the discussion above. Measurement gives a score on a scale appropriate to the metric being used. Rating determines the correlation between the raw score and the rating levels, in other words, tells us whether the score can be considered to be satisfactory. Assessment is a summary of the set of rated levels and can be seen as a way of putting together the individual ratings to give an overall picture which also reflects the relative importance of different characteristics in the light of the particular quality requirements. Final decisions are taken on the basis of the assessment.

5 ISO, EAGLES and natural language applications in practice.

It would be impossible of course to claim knowledge of all applications of the ISO standards,

even within the limited area of work on natural language. In this concluding section only those applications that came to the author's cognisance through her involvement with work in the EAGLES, ISLE and Parmenides projects are mentioned.

The ISO model of 9126/91 as extended and formalized by the first EAGLES project has been tested by application to a number of different language engineering applications. Within the TEMAA project it was applied to the evaluation of spelling checkers, and initial work was done on quality models for grammar checkers and translation memory systems. As part of the EAGLES project itself, a number of projects in the general field of information retrieval were asked to apply the framework, and produced, in those cases where the project included a substantial evaluation component, encouraging results. The second EAGLES project was, for the evaluation group, essentially a consolidation and dissemination project, where an attempt was made to encourage use of earlier results. During this time, the model was also applied in the context of the ARISE project, which developed a prototype system whereby information on railway timetables could be obtained through spoken dialogue. Similarly, an Australian manufacturer of speech software used the framework to evaluate a spoken language dialogue system. Case studies undertaken in the context of post-graduate work have applied the ISO/EAGLES methodology to the evaluation of dictation systems, grammar checkers and terminology extraction tools. One part of the ISLE project, now coming to an end, has been applying the methodology to the construction of a large scale quality model of machine translation systems. Many of the results of this work can be consulted by looking at the EAGLES and ISLE web sites.

Recently, work has begun on the Parmenides project. This project is concerned with ontology based semantic mining of information from web based documents, with a special interest in keeping track of information which changes over time. Evaluation plays an important role in the project. Three separate user groups are supplying the basis for case studies. At the time of writing, user requirements are being defined, which will be translated into quality requirements for the software to be developed within the project and which will serve as the basis for the quality models to be used in on-going and final evaluation.

6 Conclusion.

The workshop for which this paper has been written addresses the question of whether there is anything that can be shared between evaluations. The answer which I hope to have made convincing is that one thing which can be shared is a way of thinking about how evaluations should be designed and carried out. Adhering to an acknowledged standard in the construction of quality models and in developing the process of a specific evaluation can only make it easier to share more detailed aspects of evaluation and provides a common framework for discussion of such issues as metrics and their validity.

References.

Blasband, M. 1999. *Practice of Validation: the ARISE Application of the EAGLES Framework*. EELS (European Evaluation of Language Systems) Conference, Hoevelaken, The Netherlands.

EAGLES Evaluation Working Group. 1996. *EAGLES Evaluation of Natural Lnaguage Processing* Systems. Final Report, Center for Sprogteknologi, Copenhagen, Denmark.

Hovy, E, King, M and Popescu-Belis, A. 2002. *Computer-aided Specification of Quality Models for MT Evaluation*. Third International Conference on Language Resources and Evaluation (LREC).

Hovy, E, King, M and Popescu-Belis, A. 2003. *Principles of Context Based Machine Translation Evaluation*. ISLE report.

ISO/IEC 9126-1:2001 *Software engineering – product quality – Part 1: Quality Model*. Geneva, International Organization for Standardization and International Electrotechnical Commission.

ISO/IEC DTR 9126-2 (in preparation): *Software engineering – product quality – Part 2: External metrics*. . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC CD TR 9126-3 (in preparation): *Software engineering – product quality – Part 3: Internal metrics.* . Geneva, International Organization for

Standardization and International Electrotechnical Commission

ISO/IEC CD 9126-4 (in preparation): *Software engineering – product quality – Part 4: Quality in use metrics*. . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC CD 9126-30 (in preparation): *Software engineering – Software product quality requirements and evaluation – Part 30: Quality metrics* – Metrics reference model and guide. . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-1:1999 *Information technology – Software product evaluation – Part 1: General Overview.* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-2:2000– *Software engineering - product evaluation – Part 2: Planning and Management.* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-3:2000– *Software engineering - product evaluation – Part 3: Process for developers.* . Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-5:1998 *Information technology – Software product evaluation – Part 5: Process for evaluators* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-4:1999– *Software engineering - product evaluation – Part 4: Process for acquirers* Geneva, International Organization for Standardization and International Electrotechnical Commission

ISO/IEC 14598-6:2001– *Software engineering - product evaluation – Part 6: Documentation of evaluation modules* Geneva, International Organization for Standardization and International Electrotechnical Commission

King, M. 1996. *Evaluating Natural Language Processing Systems*. Communications of the Association for Computing Machinery (CACM), Vol. 39, Number 1.

Popescu-Belis, A. 1999. *Evaluation of natural anguage processing systems: a model for coherence verification of quality measures*. M. Blasband and P. Paroubek, eds, *A Blueprint for a General Infrastructure for Natural Language Processing Systems Evaluation Using Semi-Automatic Quantitative Approach Black Box Approach in a Multilingual Environment*. ELSE project. (Evaluation in Speech and Language Engineering).

Sparck-Jones, K. and Galliers J.R. 1996. *Evaluating Natural Language Processing Systems:An Analysis and Review*. Lecture Notes in Artificial Intelligence 1083. Springer-Verlag.