Standards work related to evaluation

Maghi King

1. A little history.

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) together form the specialized system for worldwide standardization. International Standards are developed by technical committees, whose membership comes from national bodies who are members of ISO or from IEC participants. ISO and IEC committees collaborate in fields of mutual interest.

General information about ISO, and about the two series of ISO standards which relate to management, ISO 9000 and ISO 14000, can be found at

http://www.iso.ch

The 9000 series primarily deals with quality assurance, the 14000 series with management and the environment.

Information on how to order ISO documents can be found at the same address. Throughout this section, quotations from ISO documents are given in italics.

An important standard pertaining to evaluation is ISO/IEC 9126, which was prepared by Joint Technical Committee JTC 1, Information technology.

The first edition of this standard, entitled "Information technology - Software product evaluation - Quality characteristics and guidelines for their use" was published in 1991.

As its title implies, this standard was mainly concerned with stipulating a set of quality characteristics for software, worked out on the basis of the general definition of quality that was used in ISO 8402. The definition of quality is accepted for all kinds of products and services. It starts from the user's needs.

"The totality of features and characteristics of a product or service that bears on its ability to satisfy stated or implied needs" (ISO 8402: 1986, note 1).

On the grounds that a set of definitions given only as an exercise in terminology would not provide sufficient support to those involved in assessing software quality, a description on how to proceed with evaluating the quality of a software product was also included.

It was acknowledged in the standard that evaluating product quality in practice required characteristics beyond the set given, and also required the development of metrics associated with each of the quality characteristics. However, the state of the art did not permit standardization in those areas, and rather than wait an indefinitely long period of time for the necessary enhancements, it was decided to issue the 1991 version to harmonise further development.

In 1994 it was felt that other standards being produced in the area of product quality evaluation necessitated the revision of 9126. The revision has resulted in a series of

documents. The **quality model** and documents on **metrics** pertaining to it form part of the 9000 series. The **process of evaluation** has been separated out and is the topic of a series of documents in the 14000 series.

That revision is now almost complete, at least for the part which directly concerns the definition of quality. The draft of ISO/IEC 9126 Part 1, the quality model is, at the time of writing, at the Final Committee Draft stage. No major changes are now expected.

Similarly, a new standard ISO/IEC 14598-1, which gives a general overview of the process of evaluation, is very close to publication as an international standard. The other documents in the 9126 and 14598 series are still at the working draft stage, and are not reported on here.

Both the 1991 version and the new versions are considered in more detail below. The 1991 version is referred to as ISO 9126, 1991, the new versions as ISO 9126, nd (for "new draft", since the date of publication is not yet known).

2. EAGLES and ISO/IEC.

The first phase of EAGLES work started in 1993. A primary goal of the initiative was standardization in the language engineering area. Naturally enough, what could or should be standardized varied from one working group to another. For the Evaluation working group, where it was felt that evaluation methods and techniques were at an early stage of development, the aim was to produce a way of thinking about evaluation rather than a set of recipes for the evaluation of particular types of systems. In particular, there was substantial agreement within the group that there could be no single and universal evaluation technique which could be applied to all language engineering products indifferently of the contexts in which the product would be used.

A first step therefore was to look for existing standardization work which could form a starting point for the development of a methodology for ***evaluation design***: a way of thinking about evaluation which could be applied to the construction of any specific evaluation, and which, since it would be common to all evaluations of language engineering products, would provide a de facto standard at an appropriate level of abstraction, permitting the particularities of specific evaluations to be taken into account within a standardized framework.

Indeed, even though work concentrated on commercially available or near-to-market products, it was intended that the principles of evaluation design worked out within the project should be much more widely applicable, and should be capable of being used for evaluation at any point of the product's life cycle, from initial project proposal through development to commercialisation.

From this perspective, ISO/IEC 9126, 1991 was of considerable interest: it fitted almost exactly with what the group was looking for. Furthermore, it was part of the mandate of the EAGLES group that users needs and requirements should play a major role in the framework to be devised. This fitted in very closely with the ISO definition of quality, recalled here:

"The totality of features and characteristics of a product or service that bears on its ability to satisfy stated or implied needs" (ISO 8402: 1986, note 1).

ISO/IEC 9126, 1991 was therefore very influential on the work of the group, and a great deal of effort was invested into first deciding what modifications and extensions would be necessary in order to apply the standards and guidelines in practice to the evaluation of language engineering systems, secondly into producing a formal version of a model of quality.

The first exercise involved defining quality characteristics and sub-characteristics for a number of different classes of systems. The characteristics for spelling checkers were worked out in some detail, a fairly substantial check-list for translation memory systems was produced, and work on grammar checkers was started. The work on spelling checkers and grammar checkers was mainly carried out in the framework of an LRE project, TEMAA, which carried the work on spelling checkers further by defining metrics for the quality sub-characteristics which had been identified. An account of that work can be found in section XXX of this report, and in the TEMAA final report.

Formalisation involved formal description of the quality characteristic hierarchy in terms of a feature structure of the type familiar from work in computational linguistics. Additional work on metrics and on automation within the TEMAA project allowed a prototype Evaluator's Workbench to be developed. Within the workbench environment, some measurements could be carried out (semi)-automatically, and a report could be automatically generated which assessed the suitability of a particular system in the light of the specific needs of a user or of a class of users. This latter was made possible by using the same descriptive tools for the description of users as those used for the description of systems, and by providing mechanisms for reflecting the relative importance of particular sub-characteristics for specific users. That work too is described in more detail elsewhere in this report (XXX).

The second round of EAGLES Evaluation work started in 1996 and is now drawing to a close. It was seen primarily as a consolidation and dissemination effort, with no new work on developing the EAGLES framework being undertaken within the group itself. During this phase, the group has been fortunate enough to have been able to enter into direct content with the Document Editor of the new drafts of ISO/IEC 9126 and of ISO/IEC14598-1. The draft of 9126 was presented in an Evaluation Group workshop in November of 1997. It was particularly pleasing to be able to notice a convergence of ideas, especially in the area of the importance of metrics. Subsequent examination of the draft of ISO/IEC 14598-1 has confirmed the convergence of ideas.

3. ISO/IEC 9126. First edition, 1991.

Since later revision has resulted in a division of the subject matter, discussion of ISO 9126, 1991 is here placed under two separate headings, even though both topics are covered in the same document in the 1991 standard.

The account is intended to be a brief summary of the documents in question, with occasional commentary touching on the relationship between EAGLES work and ISO. The commentary is of course entirely the responsibility of the EAGLES group, and in no way reflects ISO policy.

The Quality Model.

It has already been mentioned that the quality model set out in ISO/IEC 9126 is based on a general definition of quality, quoted above, which is intended to be applicable to any product or service. The model in 9126 is therefore a specialization of the generic model, intended as a quality model specifically of software product. Quality is seen in general as a composite of a set of quality characteristics. Relevant quality characteristics must be chosen and defined in order to produce a specialized quality model.

The requirements for choosing the quality characteristics set out in 9126 were as follows:

- to cover together all aspects of software quality resulting from the ISO quality definition
- to describe the product quality with a minimum of overlap
- to be as close as possible to the established terminology
- to form a set of not more than six to eight characteristics for reasons of clarity and handling
- to identify areas of attributes of software products for further refinement.

We recall that the definition of quality on which 9126 is based is that of ISO 8402: 1986:

"The totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs".

It is perhaps worth underlining here once again that this definition fits in very closely with the mandate given to the EAGLES Evaluation group to ensure that user needs play a central role in evaluation. Even though evaluation may be carried out at many different points in a product's life-cycle, and by many different people, thus giving rise to what 9126 calls different view-points on evaluation, the ultimate objective is always the satisfaction of user needs. Evaluation during development, for example, is aimed at predicting whether a product will ultimately satisfy user needs or not.

Six quality characteristics of software were stipulated in the standard: **functionality, reliability, usability, efficiency, maintainability, and portability**. We shall not give the detailed definitions here.

It is important to note that each of these characteristics was perceived to be the top level of a hierarchy of sub-characteristics. An annexe to the standard, Annexe A, whose status was informative rather than normative, gave examples of how each characteristic could be broken down into a set of sub-characteristics, each of which, could in its turn be further broken down. There is no claim that the sub-characteristics of Annexe A and their organisation constitute the only possible model of quality which can be derived from following the standard. Rather,

"The key point is that there should be a quality model to at least the subcharacteristic level for a software product, not that it should be of the precise form described in this annex." (ISO/IEC 9126, Annex A).

The guidelines contained in the body of the document also point out that the importance of each quality characteristic will vary, depending on the class of software.

"For example, reliability is most important for a mission critical system software, efficiency is most important for a time critical real time system software, and usability is most important for an interactive end user software." (ISO/IEC 9126: 1991, 5.1 Usage).

We have already mentioned that 9126 points out that there may be different views of software quality. Those discussed in the document itself are those of the user (who may be an end-user in the conventional sense of end-user, but may also be an operator, a recipient of the results of the software, or even a developer or maintainer of the software: the essential point being that the user uses the system to perform a specific function), the developer or the manager. It is emphasized that the developer may use different metrics for some characteristic than the user. For example, the user may think of efficiency in terms of response time, while the developer, at some stage of development, may not be able directly to measure response time. But since he is by necessity ultimately interested in the same quality characteristics as the user, he will use other metrics, such as path length and access or waiting time to predictively measure the same characteristic.

"Generally speaking, metrics applying to the external interface of a product are replaced by those applying to its structure". (ISO/IEC 91126: 1991, 5.2.2 Developer's view.)

We can summarize the quality model set out in 9126 by saying that a set of quality characteristics are stipulated, which can, and should be further broken down into sub-characteristics. The hierarchical structure thus obtained for some class of software product is a model of quality for that product. The quality characteristics, and especially the subcharacteristics given in Annex A are not rigid and unchangeable: their primary purpose is to serve as a check-list, guiding the evaluator in his attempt to decide and define what characteristics contribute to quality and therefore should be measured when carrying out an evaluation.

The Evaluation process model.

The evaluation process model given in 9126 is part of the guidelines for use of the quality characteristics. Three stages of the process are distinguished,

- quality requirements definition
- evaluation preparation
- evaluation procedure.

The evaluation process is conceived of 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.

Quality requirements definition involves setting up a model of quality for the product in question. The model defined will capture the stated or implied needs of the user, and will express the demands made by the environment upon the software produced. Requirements for system components may be derived from requirements for the whole system, but, typically, different requirements will be made on different components. The quality requirements are expressed in terms of quality characteristics and sub-characteristics.

Evaluation preparation involves three sub-phases:

- quality metrics selection
- rating levels definition
- assessment criteria definition

Quality 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. However, metrics used during the development phase should correlate to the metrics used when evaluating from the user view, since ultimately only the user view matters.

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. **Rating levels definition** 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 obtained 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 the system does not perform satisfactorily 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 essential characteristic, it is its performance with respect to all the other characteristics which will determine whether the system is acceptable or not.

Assessment criteria definition involves defining a procedure for summarizing the results of the evaluation of the different characteristics, using for example decision tables or weighted averages.

Note that quality metrics selection, rating levels definition and assessment criteria definition all form part of the preparation of the evaluation, and are done before any measurement actually takes place.

One might comment that there are obvious good reasons for insisting that the three sub-phases above are part of the preparation. It is only too easy for the evaluator to be influenced by the results of the measurement, and to change his criteria for acceptability. Setting out those criteria before the measurement is done at least helps to minimize this danger.

The last stage is the evaluation procedure itself, broken down into;

- measurement
- rating
- assessment

These steps are intuitively straightforward in light of the 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. Assessment is a summary of the set of rated levels. On the basis of this assessment, a final managerial decision is taken based on management criteria.

It is perhaps worth noting that all the steps above are mirrored rather faithfully in the prototype Evaluator's Workbench produced by the TEMAA project, and reported on in Section XXX.

Another point is worth making before turning to the later versions of the ISO standard. The overall perspective of the ISO standard is that of software development: in the statement of scope we are told that the Standard is intended for those associated with "acquisition, development, use, support, maintenance or audit of software." This is a viewpoint quite different to that of the comparative evaluations carried out in the framework of technology evaluation, such as the American programmes in various fields and the more recent comparative evaluation efforts in the Francophone world. (See Appendix XXX for more information).

This might lead the reader to believe that the evaluation of core technology and the sort of product or system evaluation presupposed by the ISO standards are fundamentally different. The EAGLES belief is that this is not so. The procedures set out in the ISO documents as well as the way of thinking reflected in the proposed ways of constructing definitions of models of quality are generic, and apply to all manner of evaluation. Indeed, the more recent ISO documents suggest that they may be useful even outside software evaluation and applicable to any complex product. In other words, the ISO documents propose a way of thinking which is part of the infrastructure basic to any evaluation design, no matter what the specifics of the particular evaluation might be.

Metrics.

Very little is said in 9126 about metrics, except that the state of the art is not sufficiently advanced for standardization work to be carried out, and that only a few generally accepted metrics exist for the quality characteristics given.

However, it is worth noticing that in this standard a metric is by definition a quantitative metric:

"3.14 ***software quality metric****: A quantitative scale and method which can be used to determine the value a feature takes for a specific software product."* (ISO/IEC 9126: 1991,3.14)

This is in contradistinction to the EAGLES proposal, where metrics are classified by the type of value they may take into facts, judgements and tests. Facts need not involve any kind of scale, and judgements are qualitative rather than quantitative, since they involve the exercise of human judgement. (For further discussion, see Section XXXX).

However, the disagreement is apparent rather than real, since 9126 elsewhere suggests that standards groups and organisations may establish their own evaluation process models for creating and validating metrics, and that

"In those cases where appropriate metrics are unavailable and cannot be developed, verbal descriptions or "rule of thumb" may sometimes be used." (ISO/IEC 9126: 1991, 5.1).

It is also interesting to note that the distinction made in the new draft ISO documents between internal and external metrics (see below) is foreshadowed in the 1991 document by the remark, when discussing the developer's view of evaluation, that the metrics used by the developer, although they should correlate with those used from the user viewpoint, will not be the same metrics. In the developer's case:
"Generally speaking, metrics applying to the external interface of a product are replaced by those applying to its structure." (ISO/IEC 9126: 1991, 5.2.2)

4. ISO/IEC standards, new draft.

What are the major changes?

Before going on to look at the recent versions of the ISO standards relating to evaluation, it might be useful to summarize the major changes.

First, it is important to notice that the basic principles have **not** changed. It is still the case that the starting point for designing an evaluation is constructing a model of quality which is based on the ISO general definition of quality quoted above. Thus it is still the case that user needs are taken as primordial.

The major changes, then, are in format and in greatly expanded working out of the content of the standard. There now two separate series of documents, one series, relating to the quality model in the 9000 series, the other series, relating to the evaluation process model, in the 14000 series.

ISO/IEC 9126: 1991 did not talk explicitly of a quality model. The new draft version explicitly specifies such a model. The quality characteristics remain unchanged, but normative sub-characteristics have been introduced, most of which are based on the illustrative subcharacteristics contained in Annex A of the 1991 standard.

A new notion "quality in use" has been introduced. Quality in use is quality as seen from the user point of view, and is super-ordinate to the six quality characteristics already defined, being a composite of them whose exact nature can only be determined by the specific requirements of a specific user in a specific environment. Quality in use is discussed in more detail in section XXXX of this report.

Metrics have moved into the foreground as an area of interest. A distinction is made between metrics relating to internal characteristics (internal metrics) of the software and metrics relating to the behaviour of the software as seen from the outside (external metrics). Internal metrics are therefore measuring characteristics of the software itself, such as the number of lines of code or the number of function calls. External metrics relate to the behaviour of the software when it is run; the two internal metrics above might well contribute to response time, which is an external metric. Although there is no necessary one to one relationship between internal and external characteristics and their associated metrics, internal metrics are predictors of external metrics, just as external metrics are predictors for quality in use. Documents on each of these two types of metrics are in preparation. Recently, a work item on

metrics for quality in use has been added. This gives us the following documents in the 9126 series:

9126-1: Quality model
9126-2: External metrics
9126-3: Internal metrics
9126-4: Quality in use metrics.

9126-1 is close to publication as an International Standard. The other documents in the series are in preparation.

The evaluation process model has been removed from the 9126 series and is now part of the 14000 series. ISO/IEC 14598 consists of the following parts under the general title Information Technology - Software product evaluation:

14598-1: General overview
14598-2: Planning and management
14598-3: Process for developers
14598-4: Process for acquirers
14598-5: Process for evaluators
14598-6: Documentation of evaluation modules.

The discussion in the present draft of this document is based on the current version of 14598-1, which is close to publication as an International Standard.

ISO/IEC 9126-1: Final Committee Draft, nd: Quality model.

The normative part of ISO/IEC 9126-1, nd is the definition of a quality model. The model distinguishes internal quality, external quality and quality in use.

It specifies six quality characteristics (the same six as those specified in 9126, 1991) for internal and external quality. The quality characteristics are broken down into subcharacteristics which now are an integral part of the normative work. Quality in use is broken down into four characteristics which are the combined effect of the software quality characteristics from the user's point of view.

The intended use of 9126: nd is very wide indeed. The characteristics defined are meant to be applicable to any kind of software, and also to provide a consistent terminology for software quality. Their chief purpose is to provide the framework for specifying quality requirements. The intended users of 9126: nd include developers, acquirers, quality assurance staff and independent evaluators. Examples of uses of the quality model include:

- validation of the completeness of a requirements definition
- identification of software requirements
- identification of software design objectives
- identification of software testing objectives
- identification of quality assurance criteria
- identification of user acceptance criteria for a completed software product

IS/IEC 9126-1: nd can also be used in conjunction with other ISO standards in a wide variety of tasks, including software process assessment, definition, review, verification and validation of quality requirements during software lifecycle and quality assurance processes.

A quality model is described which explains the relationship between different approaches to quality. The breakdown into quality characteristics and subcharacteristics constitutes a specific implementation of the generic quality model.

A distinction is made between internal measures, which are typically static measures of intermediate products (by which is meant specifications, source code etc.) and external measures, which typically involve measures of the behaviour of the code when executed. Different approaches to quality then go in a chain from process quality, which influences internal quality, which in turn influences external quality which in turn influences quality in use. Seen from the opposite end of the chain, quality in use depends on external quality, which depends on internal quality, which depends on process quality. This question of approaches to quality, and of how different kinds of quality relate to one another is a particular problem for usability and is therefore discussed in considerably more detail in section XXXX of this report.

Standing outside the chain, goal quality is the necessary and sufficient quality which reflects real user needs. Goal quality is not necessarily perfect quality, but the quality which allows the user to achieve his goals. It is not always possible to define goal quality completely before development starts, partly because real user needs are not always consciously known or stateable at that point, partly because user needs may change and develop during the development process. The case study in section XXXX (Marc: ARISE) offers a practical illustration of this point: there, the initiators of the system design assumed a particular user profile, and therefore a particular set of user needs. In practice, evaluation of the system through scenario testing (see EAGLES I Final Report 1996 for a description of scenario testing) showed the profile to have been misjudged, and the user needs therefore to have been wrongly defined.

The item to be evaluated differs according to the approach to quality. For process quality, it is the process itself. For internal and external quality, it is the software product. For quality in use, it is the effect of the software product.

This latter is rather important: to quote the new draft:

"Software never runs alone, but always as a part of a larger system consisting of other software products with which it has interfaces, hardware, human operators, and work flows...Quality in use (the capability of a product to meet stated and implied needs) can be measured by the extent to which a product used by specified users meets their needs to achieve specified goals with effectiveness, productivity, safety and satisfaction." (ISO/IEC 9126-1: nd).

This is very reminiscent of what earlier EAGLES work called a set-up (based on a term used by Karen Sparck Jones in making essentially the same point as the quotation above). There, it was also pointed out that one set-up may be embedded inside another, thus leading to an important distinction, also made in 9126, between evaluation of a software product and evaluation of the system in which it is executed. ISO/IEC 9126-1 gives a clear example of

how where the boundary of the system is considered to be may change depending on the purposes of the evaluation and on who the users are taken to be:

"For example, if the users of an aircraft with a computer-based flight control system are taken to be the passengers, then the system on which they depend includes the flight crew, the airframe, and the hardware and software in the flight control system, whereas if the flight crew are taken to be the users, then the system upon which they depend consists only of the airframe and the flight control system." (ISO/IEC 9126-1, nd, 5.3).

9126 summarizes the use of a quality model in evaluation as follows:

"Software product quality should be evaluated using a defined quality model. The quality model should be used when setting quality goals for software products and intermediate products, Both software quality and quality in use should be decomposed into a quality model composed of characteristics and subcharacteristics which can be used as a checklist of issues relating to quality." (ISO/IEC 9126-1, nd, 5.4).

Clauses 6 and 7 of 9126-1, nd, define a hierarchical quality model for software quality and quality in use, although it is noted that other ways of categorising quality may be more appropriate in particular circumstances. Despite this last qualification, it is noted elsewhere in the document that the model given is the default model: other models should not be used unless there is good reason to do so.

The quality characteristics.

We shall not give here the full definition of each of the quality characteristics and their subcharacteristics, restricting ourselves to a simple naming exercise, a few sample definitions and some comments. The reader is referred to the ISO document for full definitions. The series of tables below set out for each quality characteristic the sub-quality characteristics into which it is broken down. Quotations from ISO 9126-1, nd, are, as usual, in italics.

Functionality	
	Suitability
	Accuracy
	Interoperability
	Security
	Compliance

Suitability: *The capability of the software to provide an appropriate set of functions for specified tasks and user objectives.*

Accuracy: *The capability of the software product to provide the right or agreed results or effects.*

Reliability	
	Maturity
	Fault tolerance
	Recoverability
	Compliance

In the context of reliability it is worth noticing that further characteristics are sometimes introduced into the standard, other than those directly given as defined quality characteristics or subcharacteristics. An example here is Availability, which is *the capability of the software product to be in a state to perform a required function at a given point in time, under stated conditions of use.* It is said to be a combination of maturity (which governs the frequency of failure), fault tolerance and recoverability (which governs the length of down time following each failure). The EAGLES version of a quality model as a formalized hierarchy of attributes and sub-attributes models this view.

Usability	
	Understandability
	Learnability
	Operability
	Attractiveness
	Compliance

The notes given on the various definitions of this quality characteristic and its subcharacteristics make several interesting points. First, they make it clear that quality sub(characteristics) are inter-dependent: for example, some aspects of functionality, reliability and of efficiency will clearly affect usability, but are deliberately excluded from mention under usability in the interests of keeping the quality model tidy and well structured. They will come into play when considering the super-ordinate characteristic, quality in use. Similarly, aspects of suitability, changeability, adaptability and installability may affect the subcharacteristic operability. Secondly, the notes emphasize that usability issues affect all the different kinds of users: *users may include operators, and users and indirect users who are under the influence of or dependent on the use of the software. Usability should address all of the different user environments that the software may affect, which may include preparation for usage and evaluation of results.*

Efficiency	
	Time behaviour
	Resource utilisation
	Compliance

Maintainability	
	Analysability
	Changeability
	Stability
	Testability
	Compliance

In earlier EAGLES work, it was argued that for language technology software, the ability to customize the software to a particular user's needs was of extreme importance, and that there was no natural place in the ISO 9126, 1991 definition of the quality characteristics where customizability would fit. For that reason, customizability was added as a seventh quality characteristic. This extension is no longer necessary in view of a note on the maintainability characteristic as a whole, which specifically includes amongst possible modifications adaptation of the software to meet user requirements and of a further note to the subcharacteristic changeability which points out that where the software is to be modified by the end user, changeability may affect operability.

Portability	
	Adaptability
	Installability
	Co-existence
	Replaceability
	Compliance

Co-existence is the capability of co-existing with other independent software in the same environment. Replaceability is the capability of the software to be used in place of another specified software product for the same purpose in the same environment, for example when a software is upgraded.

As we have already mentioned, there is one super-ordinate quality characteristic, quality in use. This will not be discussed here, since it is one of the main topics of section XXXX.

ISO/IEC 14598-1: FCD Information Technology - Software Product Evaluation - Part 1: General Overview.

14598-1, as its title implies, is mainly concerned with giving an overview of how all the different 9126 and 14598 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 the construction of the quality model.

The scope of the standard is intended to very wide. Indeed, a note states that *"Much of the guidance in ISO/IEC 14598 is not specific to software, but is also applicable to other complex products."* (ISO/IEC 14598, nd, 1. Scope.) However, in order to provide more detailed support for those involved in software evaluation, 14598 expands greatly on the notion of

"viewpoints" already mentioned in ISO 9126, 1991, by providing separate documents for certain classes of users:

5.2.1 ***Process for developers***

ISO/IEC 14598-3 should be used by organisations that are planning to develop a new product or enhance an existing product and intending to perform product evaluation using members of its own technical staff. It focuses on the use of those indicators that can predict end product quality by measuring intermediate products during the life cycle.

5.2.2 ***Process for acquirers***

ISO/IEC 14598-4 should be used by organisations that are planning to acquire or reuse an existing or pre-developed software product. It can be applied for the purposes of deciding on the acceptance of the product or for selecting a product from among alternative products. (A product may be self contained, a part of a system, or it may be part of a larger product.)

5.2.3 ***Process for evaluators***

ISO/IEC 14598-5 should be used by evaluators carrying out an independent assessment of a software product. This evaluation could be performed at the request of either a developer, acquirer or some third parity. This part is intended for those who perform independent evaluation. Often they work for third party organisations.

All these classes of users (of evaluation) will make use of ISO 9126. They are further supported by the second half of ISO 14598-1, which sets out a generic picture of the process of evaluation, and by two further documents.

ISO/IEC 14598-2 Planning and management is related to *planning and management of a software evaluation process and associated activities, including development, acquisition, standardisation, control, transfer and feedback of evaluation expertise within the organisation.* (ISO 14598, nd, 5.3.1).

ISO/IEC 14598-6 provides guidance for documenting evaluation modules. These modules contain the specification of the quality model to be used, together with data associated with the application of the metrics. They also contain information about how it was planned to apply the model and about its actual application. Evaluation can sometimes be re-used: for each evaluation appropriate evaluation modules are then selected. In other cases it may be necessary to develop new evaluation modules. 14598-6 is intended for use by organisations producing new modules. It is perhaps worth noting that in the case of human language technology, very few quality models exist: most attention so far has been paid to the development of specific metrics. It is part of the EAGLES Evaluation Working Group mission to further the development of appropriate quality models. Therefore this document, when it becomes available, is likely to be of particular importance to EAGLES future work.

The second section of ISO 14598-1, nd, is devoted to building on the evaluation process model set out as a set of guidelines in ISO 9126, 1991. The result is an expansion of that work, rather than a new version of the process model.

Evaluation process is now broken down into four main stages:

- Establish evaluation requirements
- Specify the evaluation
- Design the evaluation
- Execute the evaluation

Each of these is further broken down. Establishing the evaluation requirements involves the following three steps:

- **Establish the purpose of the evaluation:**

Examining 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:

1. decide on the acceptance of an intermediate product from a subcontractor
2. decide on the completion of a process and when to send products to the next process
3. predict or estimate end product quality
4. collect information on intermediate products in order to control and manage the process.

The inclusion of this last possibility means that the standard is meant to apply to all of what the first EAGLES report called adequacy, progress and diagnostic evaluation. Adequacy evaluation involves assessing whether a product will meet a user's needs, and can thus be assimilated to 1 above. Progress evaluation involves assessing whether a system has made progress towards some defined goal state, and can thus be assimilated to all 2, 3 and 4 above. Diagnostic evaluation involves trying to identify the cause of wrong or unexpected behaviour, and can thus be assimilated to 4 above.

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 effect of a product when it is used
- decide when to enhance or replace the product.

It is perhaps worth noting that comparative evaluation is present in only two of the above possibilities. This is important in view of a rather widespread assumption, especially in the academic community, that the only kind of evaluation that exists is comparative evaluation.

Further commentary relates the purpose of the evaluation to the specific cases of acquisition, supply, development, operation and maintenance, and shows that evaluation may be pertinent to each one of these cases. The reader is referred to the document for detailed discussion.

- **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 process quality, internal quality, external quality or qualify in use is to be evaluated. Much of the discussion is taken up again in section XXXX of this report, and so will not be considered here.

- **Specify quality model**

The quality model is, of course, to be defined using ISO 9126-1 as a guide. However, a note adds

The actual characteristics and subcharacteristics 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 subcharacteristics provide a useful checklist of issues related to quality, but other ways of categorising quality may be more appropriate in particular circumstances. (ISO 14598, nd, 7.3).

Specifying the evaluation also involves three steps:

- Select metrics
- Establish rating levels for metrics
- Establish criteria for assessment.

These are already familiar from ISO 9126, 1991, and have been described in the relevant earlier section. It is worth however noticing some additional material on metrics, which distinguishes between evaluation carried out in order to identify problems so that they can be rectified, and comparative evaluation, either against alternative products or against requirements. In the former case, "A wide range of measures can be useful for these purposes, including check lists and expert opinion. The primary requirement is that the measurements correctly identify the impact that any changes in the software may have on quality" (ISO 14598, nd, 8.1.1). Where reliable comparisons have to be made, metrics should be "more rigorous": *data from checklists and expert opinion may not be reliable when comparing products with different attributes. Allowance should be made for possible measurement errors caused by measurement tools or human error"* (ibid.). This last is an important point; many past evaluations have suffered from not being sufficiently aware of the difficulty of comparing apples and pears.

Designing the evaluation involves producing an evaluation plan, which describes the evaluation methods and the schedule of the evaluator action. The documents in the 14598 series relating to the different viewpoints (those of developers, acquirers and evaluators) will expand on this point, and the plan should be consistent with a Measurements Plan, which is to be described and discussed in the document on planning and management.

The final stage is the execution of the evaluation, which again is familiar from ISO 9126, 1991 and has already been discussed. The whole evaluation process is supported by activities for assisting evaluation by collecting information on evaluation tools and methods, developing and validating metrics and standardising evaluation process metrics and measures.

The planning and management document (ISO 14598-2) will also contain requirements and guidelines for these activities.