Standardized Acceptance Testing

Hans-Ludwig Hausen,
Fraunhofer, Schloss Birlinghoven, Sankt Augustin, Germany
hausen@gmd.de

Abstract

System evaluation and acceptance testing is considered as a critical task comprising: analysis of condition and constrains (result: evaluation requirements), specification of the evaluation (result: identification of relevant system items and associated system attributes), design of the evaluation (result: evaluation plan), conducting the evaluation (result: test and measurement reports) and finally reporting on the evaluation (result: compilation of all intermediate results). Complementary, the concept of evaluation module is introduced to allow a well-structured description of mature evaluation techniques (e.g. inspection, testing, reliability modelling) as well as the definition of their interaction, the latter being necessary to instrument an acceptance testing method by a set of coherent evaluation techniques.

1 Introduction

For many applications, most code is not devoted to implementing the primary input-output functionality but instead addresses other concerns, such as reliability, availability, responsiveness, performance, security, and manageability. Conventional programming practice requires the programmer to keep all these other ilities in mind while coding and to explicitly invoke behavior at exactly the right places to achieve them. Therefore, system quality evaluation and acceptance testing is identified by industry as an important issue for system development, distribution and application [Raea95].

A system evaluation scheme needs to be capable of dealing with any type of systems. Such systems range from *off-the-shelf* system developed for a general customer base, through projects commissioned by a single customer, to embedded systems in systems. The FIG. 1 illustrates this classification of systems and shows the parties having a direct interest in the acceptance testing.

Such a scheme must be of value both to the producers, sellers, and users of a component and to the community-at-large. It must have stability and must be trusted by all. Therefore, it must be regulated, consistent, understandable, cost effective and respected. Any scheme must be flexible, evolutionary and capable of rapid response to change. There will also be the need to harmonize the scheme with any changes in the law or of standards or regulations which impinge on systems and their use.

A practical approach should comprise: analysis of condition and constrains (result: evaluation requirements), specification of the evaluation (result: identification of relevant system items and associated system attributes), design of the evaluation (result: evaluation plan), conducting the evaluation (result: test and measurement reports) and finally reporting on the evaluation (result: compilation of all intermediate results). Complementary, the concept of evaluation module is adopted which allows a well-structured description of evaluation techniques (e.g. inspection, testing, reliability modelling) as well as the definition of their interaction, the latter being necessary to instrument the method by a set of coherent evaluation techniques.

In the following sections the methods and tools for the evaluation and assessment of systems and software processes are discussed in detail. Particular emphasis is given to the identification and selection of system characteristics and metrics as well as to the handling of evaluation methods and tools. In a situation where we have a huge amount of software metrics, the problem of identifying the right one and applying it correctly is an important issue.

2 Specified vs Actual Service

The basic task of a system evaluation process is to check for evidence that the actual service exhibited by a system is a trusted instantiation of the specified service. In other words, *Has the system been made correctly?* has to be answered. on the other hand, a system evaluation cannot ensure a correspondence between expected service (from the user's perception) and actual service - i.e.*Is it the correct system?*

A *formal* procedure like acceptance testing, therefore, must be based upon the assurance of a formal-

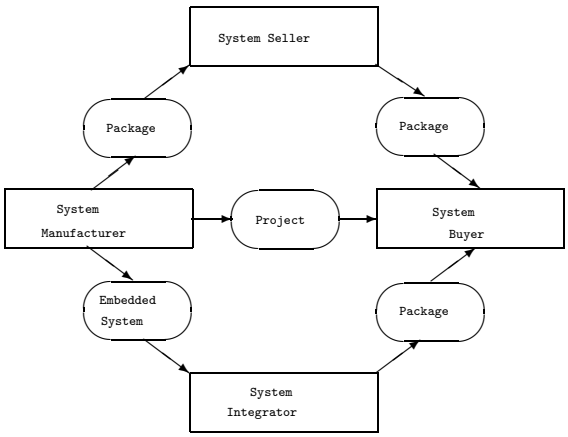

Figure 1: Views and Interests

ized description of the system. Usually, this will be contained in documents such as a requirements specification, user's manuals and instructions, design specifications etc. The main point is that the description is in a published format which does not change (at least for the period of the acceptance certificate) and can be used as a reference. It does not matter that the system does not represent the intentions of the producer correctly. The acceptance tester's job is to assure that the system is, and behaves, as it is described.

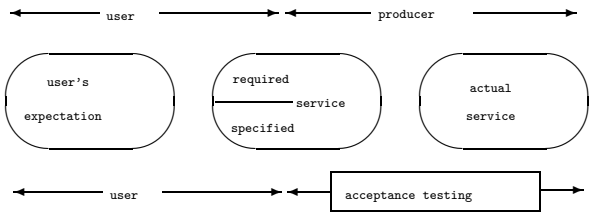

Figure 2: expectations, required, specified, and actual service

This view does, of course, present problems for users who may be expecting that a certificate will be an assurance that a system will meet their expectations. But this can only be true when the system fully meets its specifications and the specifications correspond to the expectations. Unfortunately, user expectations frequently are informal, subjective and go beyond the written descriptions (specifications) of the system (see respective discussion in [Raea95]).

The key relationships are that it must be the user's responsibility to ensure that the choice of system corresponds to his or her needs and the producer's responsibility to ensure that a system corresponds to its description.

There is one exception to the principle that the certificate should only represent correspondence between actual and specified service. That exception is where some omission or ambiguity in specification results in a system which has a potential failure mode which may be hazardous to the user or third parties. In this case, it might be expected that it was a responsibility of the certifier to identify such anomalies during the analysis of the system.

These considerations lead to two views of acceptance testing. There must exist a fundamental or basic acceptance testing which can assure conformance of the actual service of a (software) system with its specified service. Where it is necessary also to assure that the system behaves both reasonably and that its use carries no unacceptable risk, a basic acceptance testing can be extended. Extended acceptance testing would include a more rigorous analysis of the system and the additional tests, evaluations or audits which were found necessary to assure, for instance, safety in use.

It should be noted that there is no difference in the method between basic and extended acceptance testing. One is simply an extension of the other. However, some of the issues raised by extended acceptance testing (such as *is the system fit for its intended purpose?*) pose considerable difficulties.

3 Overall Approach

It is assumed that the technology of software validation, verification, testing, and measurement developed by the research community is ready to use for software acceptance testing. In consequence, our strategy is to use existing techniques to build a consistent framework and to rationalize their application. The acceptance testing framework is in two parts: the evaluation and acceptance testing model that precisely defines the notions used, and the evaluation method that describes the various steps leading to the acceptance testing of a system.

The evaluation and acceptance testing model is based on four formal sub-models which define notions introduced by the evaluation method: a system model, a software development process model, a software characteristics model and a measurement model. These models are under continuous review as the project proceeds and empirical results are fed back from the Case Studies. The system model is aimed at defining a system which is submitted for an evaluation process.

The definition consists of two steps, the identification of software parts and the classification of software parts. The software development process model identifies items of process evidence that may be useful to facilitate system measurements. In a sense, this approach tries to reconcile the two ideas of *process* acceptance testing and *system* acceptance testing.

The software characteristics model is the kernel of the evaluation and acceptance testing model. It defines the characteristics that will be assessed in a system, i.e. the acceptance testing attributes. To be of any value, the characteristics composing this model must correspond to the public notion of software quality. The major issue in producing this model is to be able to define the characteristics unambiguously. The measurement model is more difficult. One problem is to deal with the complexity of measurement and evaluation techniques. The number of applicable measures proposed in the literature is extremely large and their conditions of application vary enormously. The approach taken by the measurement model is modular; a small set of evaluation techniques and tools, that can be mastered by a specialist, is encapsulated in what is called an *evaluation module*.

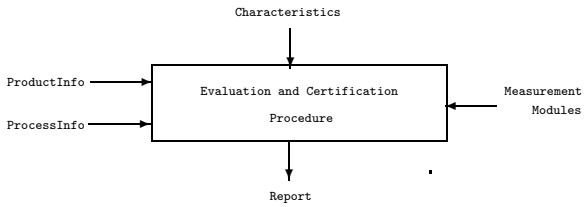

Figure 3: Input and Output of the Evaluation

The evaluation and acceptance testing method is composed of the following steps:

Identification, in the system submitted for acceptance testing, of the recognized system parts and elements of development process evidence.

Specification of the software characteristics that are to be assessed and the conformance evaluation level. The specification is based on an analysis of the descriptive system parts and will include the pass/fail criteria for the attributes of interest.

Selection and application of evaluation modules. The choice depends on the characteristics to be assessed, the available system parts and process evidence and the applicable techniques. The result of applying the evaluation modules will be a set of measurements which can be judged for conformance with the evaluation specification.

Reporting the evaluation results and evaluation of the results against the evaluation specification. This report will be the basis of the award (or refusal) of a Certificate.

4 Evaluation Procedure

What follows is a description of one possible procedure which might be refined into a scheme for assessing systems. The primary reference for acceptance testing should be a acceptance testing norm, or better an internationally agreed standard, providing details of the basis for acceptance testing, level of achievement and generally what must be done to secure a certificate. The norm should be supplemented by authoritative guides which will explain, in precise terms, exactly how the evaluation method should be applied, the attributes and characteristics that should be examined, the evaluation modules available for use and the evaluation criteria. (see respective discussion in [Raea95])

Evaluation might be undertaken by accredited test laboratories spread throughout the community. Then a certificate that is awarded will be valid anywhere in the community, irrespective of where the evaluation was done. It is crucial to the success of the scheme that application of the evaluation method by any test laboratory to a system produces consistent results.

There are a number of distinct stages in the evaluation of a system. In Fig. 4 we provide a block diagram of the evaluation process.

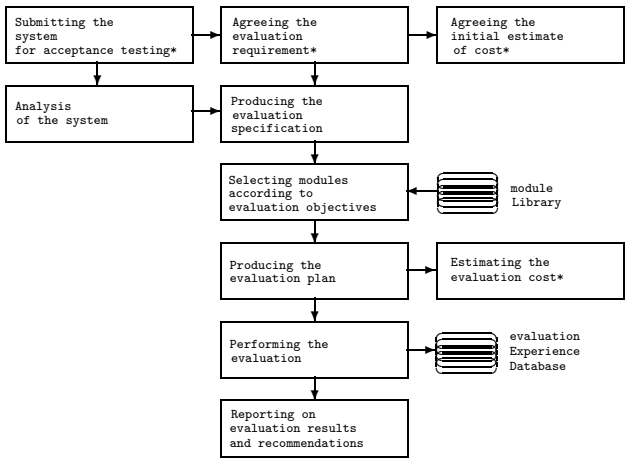

Figure 4: Stages of an evaluation Procedure

4.1 Submitting the system for acceptance testing*

The decision as to whether a system should be submitted for acceptance testing should be taken as early as possible in the life cycle of the system. If this is done right at the beginning of the design and development stages, it should be possible to 'build in' to the development, the checks and tests which the system will have to pass while it is being assessed by the Certification Agency. This should ensure the maximum chance of the system passing the evaluation, as well as minimizing the chance of extra, unexpected costs being incurred.

Early contact with the Test Laboratory to discuss the intention of submitting a system for evaluation will also help the producer to anticipate any special needs (such as particular documents or evidence which might be required) which the assessors may have. It may be that some (or even all) of the evaluation tests will have to be done 'on site', rather than at the Test Laboratory. In these cases, the tests will still be controlled by the Test Laboratory staff to ensure that the results are unbiassed. For very large, complex software projects, it may be beneficial for the producer to have a continuous, detailed association with the assessors during the whole of the development to minimize the duration and cost of the evaluation process.

When the intention to submit a system for acceptance testing is declared, (usually the decision of the producer) it will be the first task of the assessors to ensure that a minimum set of system parts (documents, code and test results) will be available to allow an evaluation of conformance to be made.

The size of the 'minimum' list will vary according to the particular circumstances of the proposed evaluation. It may be very short, such as the program object code and the user manual. These two system parts would enable an evaluation to be made as to whether the program did the things that were stated in the manual.

On the other hand, a large, complex system may need to be supported by a substantial amount of documentation, ranging from user requirements, through functional and design specification, code listings, quality and test plans, test results - to name but a few!

Extended evaluation will also require additional system parts such as risk and safety analyses and the results from beta testing.

4.2 Agreeing the evaluation requirement*

In the majority of cases, it is expected that the producer will bear the cost of acceptance testing. This will give the producer some rights as to how extensive the coverage of the certificate should be and how much the process should cost. It will be necessary, therefore, for the test laboratory and the producer to agree which software characteristics are to be assessed, the level to which they are to be assured and whether basic or extended acceptance testing is sought.

The formal record of this agreement of what will be covered in the evaluation process will be known as the evaluation REQUIREMENT. It will provide a nominal list of attributes (features, characteristics) which are to be assessed and identify the sources of data and evidence which can be used in the evaluation process.

4.3 Agreeing an initial estimate of cost*

Once the evaluation requirement is agreed, an initial cost estimate can be constructed from a knowledge of the list of items in the requirement and the work to be done.

The input to this step is the characteristics of the system which are to be assessed and the agreement on the type and level of acceptance testing. At this stage, the system has not been analyzed so a detailed knowledge of the content and quality of the system (documentation, manuals, source code etc) is not available. Only the application area and a few rough measures such as the number of documentation pages and the number of codelines and the programming language are known. Therefore, the cost estimate can only be based on the agreed acceptance testing level and the size of the system code and any previous experience of assessing similar systems. However, it should be possible to provide a reasonably accurate cost estimate for the work needed to progressing to the production of the evaluation Plan (stage 6).

4.4 Analysis of the system

It is necessary to perform an analysis of the system submitted to evaluation in order to identify the various system parts and elements of process evidence it consists of. This information is needed in order to identify which evaluation can be performed. This will be used, together with the evaluation requirements, to build the evaluation specification.

The analysis of the system consists of two phases: (i) identification of available documents, and (ii) clas-

sification of the information contained on the system and process models.

(i) Identification of submitted documents

The system submitted to evaluation consists of documents, which includes code. The first step of the system analysis consists of making a list of these documents, together with the identification of their claimed characteristics. For each document, the following information should be provided:

- title
- formalism (natural language, programming language, ...)
- claimed conformity to standard (optional, reference should be provided - language standards or development method standards should be considered)
- size (to be used for costing process)

(iia) Classification of submitted system information

The information contained in system documents belongs to the following categories:

- required service information
- specified service information
- actual service information

For each of the services, the information can be sub-classified in:

- code; data-flow, control-flow, states trace
- annotations

Of course, it is clear that most of the systems under evaluation are nor composed of documents falling strictly in the categories identified above. Some system documents contain information belonging to several classes, while the same type of information may be spread amongst several documents.

(iib) Classification of submitted process information

In order to support the evaluation of a system, the sponsor may submit documents concerning the system development process. When evaluation of these documents is to be performed, the information contained in them must be identified, so that it can be used. The information concerning the process might be classified into:

- project handbook
- quality plan
- quality reports

Some other process information might be required depending on the objectives of the evaluation and acceptance testing.

4.5 Producing the evaluation specification

A list of attributes which need to be assessed will have been derived from the evaluation Requirement, and possibly modified by the findings of the system analysis. This list represents the necessary compromise between the characteristics of the system the producer feels that it is important to assess and the more comprehensive list of characteristics that an assessor might feel to be the appropriate set of attributes which should be assessed.

The list is the basis for the evaluation SPECIFICATION which should cover:

- characteristics which are to be assessed
- sub-characteristics, which can be decomposed from the primary list (of characteristics), which will provide a link to actual measures
- a list of measurements which can be used to assess the conformance of sub-characteristics (and ultimately characteristics) with requirements, specifications, standards and legal needs
- target values for the measurements being made which will indicate whether conformance criteria are being met (pass / fail)
- class of documentation required (process or system)

The evaluation specification need not be concerned as to whether any attribute can or cannot be measured, i.e, - whether particular modules are available.

Neither should the specification be totally influenced by the wishes of the producer expressed in the evaluation requirement. If, at the system analysis stage, some aspect of the system is suspect and is deemed necessary for investigation, then this should be included in the evaluation specification. It will, of course, still be the right of the producer to withdraw the system from acceptance testing if the additional costs of the more extensive evaluation are not agreed.

4.6 Selecting evaluation modules according to evaluation objectives

The input to this stage will be the evaluation Specification. The objective is to attach to each of the bottom level acceptance testing attributes one or more measurement techniques or 'modules'. The output from this stage will, therefore, be a list of modules which are to be applied to perform the evaluation of each attribute.

The selection process itself requires that the module library be searched for modules which will be useful in the evaluation of each attribute for the target system. This directly implies two criteria to be applied in the selection of modules: First, the module must be known (and proven) to be useful in the evaluation of the attribute it is to be used for. Secondly, the module must be applicable to the system part it is to be used on. For example, many 3GL software metrics cannot be applied to object oriented or rule based software.

4.7 Producing the evaluation plan

In the previous stage, possible evaluation modules have been identified and associated with system parts. However, this set of modules may not be optimal for carrying out the evaluation. Some modules may be redundant and some modules may be missing. It must be decided whether new modules must be developed or whether missing modules can be substituted by a combination of existing modules. The purpose of this step is to make the final planning of modules for the evaluation. The planning will be done in order to optimize the coverage of evaluation and the cost of carrying through the evaluation.

4.8 Estimating the evaluation cost*

The evaluation plan includes the list of modules to be applied. Each module includes information from which the cost of its application can be derived. Hence, it is easy to calculate the total cost of the evaluation. However, the test laboratories act in a competitive market and therefore the actual price of the evaluation may differ substantially from the calculated cost. Furthermore, in some cases, the cost may also include the whole, or part of the development cost of a new module.

4.9 Performing the evaluation

The implementation of the evaluation plan means applying the modules on the related system parts and collecting for each of them the application results. The output will be a collection of measurement reports resulting from the application of the modules.

This step consists in:

(i) planning and managing the evaluation project

These are the usual activities to be performed at the beginning and during a project, namely: identifying the evaluation activities, identifying resources (human, tools, ...), allocating resources or activities and scheduling, reporting progress.

(ii) performing the measurements on the target system parts

Measurements can be manual, computer aided (e.g., using a check list manager for applying check lists), or automatic (e.g., measuring the cyclomatic number in a source code system using a static analyzer).

The main task is to collect the measurement result and also to keep any information (measurement data) about the measured system part, that could be helpful for pass/fail decision to be taken.

These data can be figures, diagrams, parts of documentation etc.

(iii) producing the measurement reports

This consists, for all modules, in collecting and synthesizing measurement data and results in order to produce the report resulting from the application of the module. The structure of this report is pre-defined in the module document.

4.10 Reporting on evaluation results and recommendations

(i) Making the decision

The actual module application results contained in the measurement reports, are to be compared to expected results specified in the evaluation specifications. Technical expertise might be required to make the pass/fail decision. This expertise has been gained from previous experience in system evaluation and from experience in the industrial sector the system belongs to.

(ii) General reporting and recommendations

The general report is a synthesis of all measurement reports. If the decision is *pass* for all modules, this report should only recapitulate the evaluation specifications, and the actual results.

If the decision is *fail*, the measurement result that drove this decision should be highlighted. Recommendations can be made to the sponsor of the evaluation in the sense of improvement of the submitted system.

In any case, all documents produced during the evaluation, measurement results and data, should be referenced in this document in order to be able to control the correct application of evaluation procedures

and the suitability of decisions taken.

(iii) Capitalizing of the experience
Experience gained from the current evaluation process must be stored. Running a system evaluation provides information that can help to precise the cost of module application, to improve module documents, to identify needs for new evaluation techniques.

This step only concerns the acceptance testing laboratory.

5 Evaluation Characteristics

The evaluation characteristics represent a selection of properties or attributes of a system. The evaluation process measures and assesses these attributes and the certificate is a statement of the extent to which the attributes are present in the system.

The choice of evaluation attributes is important for the acceptance of the idea of software acceptance testing. Many attributes can be suggested and many pros and cons can be given for each. However, a number of basic requirements can be formulated and will be helpful for the selection process. (see respective discussion in [Raea95])

The evaluation attributes must be relevant for the user (buyer) of the system, i.e. they must tell something important about the software. The evaluation attributes must be unambiguously defined and intuitively easy to understand; i.e. they must be meaningful to the users. The evaluation attributes must be measurable and measures must be reproducible, i.e. the evaluation must be based on scientific principles. Workmanship, Correctness, Reliability and Efficiency might be considered as core criteria, but also the whole Model or McCall[McCa77] might be necessary.

Internationally relevant is the quality standard ISO/IEC 9126 [ISO9126]. Here we get following six characteristics:

- FUNCTIONALITY a set of attributes that bear on the existence of a set of functions and their specified properties. The functions are those that satisfy stated or implied needs.
- RELIABILITYa set of attributes that bear on the capability of software to maintain its level of performance under stated conditions for a stated period of time.
- USABILITYa set of attributes that bear on the effort for use and on the individual evaluation of such use by a stated or implied set of users.
- EFFICIENCY a set of attributes that bear on the relationship between the level of performance of the software and the amount of resources used, under stated conditions.
- MAINTAINABILITY a set of attributes that bear on the effort needed to make specified modifications.
- PORTABILITY a set of attributes that bear on the ability of software to be transferred from one environment to another.

A further decomposition of these six characteristics to sub-characteristics and metrics is given in the quality model section of [Raea95]

6 Evaluation Modules

The quality model describes how measurement and evaluation techniques are organized. The goal is to allow an effective selection and application of these techniques. The state of the art relative to them is extremely large. In order to cope with this complexity, the proposed Measurement Model is modular: A set of complementary measurement and evaluation techniques are encapsulated in an element that is called an *evaluation module*.

An evaluation module can be considered as a blackbox that when suitable inputs are applied, returns one or more measurements which can be used to assess the conformance of a system attribute with its specification. An evaluation module should be independent of system type, unambiguously defined and produce consistent results. The format and detail necessary for describing a module is given in appendix A.

The implementation of a module, in a concrete case, is a procedure that takes as input elements of the system or evidence from the process and produces a value on a specified scale (nominal, ordinal, interval, ratio or absolute) and a report providing information on the practical module application.

The evaluation of a system (in order to certify it) is performed by applying several evaluation modules. The global pass/fail decision is obtained by evaluating the measurements from the modules with respect to the system and evaluation specifications.

A worthwhile way of considering a module is that it primarily measures a property of a given system. This can be compared to measuring physical characteristics such as length, humidity, electric current etc.

The evaluation criteria, however, are not only physical. They are derived also from technical, legal or commercial considerations. The selected evaluation attributes strongly influence the set of necessary evaluation modules. Obviously, each module must be associated with at least one attribute and conversely each attribute must be assessed by modules. The choice of modules for assessing a particular piece of software will be based on the type of available information. For

example, the programming language used may restrict the number of available tools for static analysis.

On the other hand, a module might not be explicitly related to any higher level quality characteristics. Its result may be used in the evaluation of various characteristics. However, the definition of modules using other module results will provide some structuring elements.

It may be that no complete set of modules exists that would completely assess to the required level for a particular system. In this situation there may be a case for constructing a new module or rejecting acceptance testing of the system.

7 Summary and Conclusions

In summary the Evaluation Procedure presented is *adaptable to circumstances* of any testing laboratory (available personnel, evaluation methods and tools), and *flexible to relevant standards* of system type dependent quality requirements and of software engineering processes, and to legal or contract issues. One prime objective for proposing the acceptance testing procedure is to ensure that an evaluation process is pragmatic and effective. In order to achieve this goal it is necessary to reflect
- experience with evaluation modules and the module library,
- appropriateness of levels and software characteristics,
- appropriateness of system representation,
- appropriateness of process representation,
- calculation of actual costs in order to improve cost estimates,
- appropriateness of the evaluation method.

Running a system evaluation provides information that might help to estimate the cost of an evaluation module application, to improve module documents, to identify needs for new evaluation techniques. The experience gained should be stored in a data base in order to make them available for further investigations which could lead to an improvement of evaluation procedures or particular techniques.

References

[—] References and further Readings.

[Abr00] Pekka Abrahamsson and Timo Jokela *Development of Management Commitment to Software Process Improvement* Proceedings of IRIS 23. Laboratorium for Interaction Technology, University of Trollhttan Uddevalla, 2000. L. Svensson, U. Snis, C. Srensen, H. Fgerlind, T. Lindroth, M. Magnusson, C. stlund (eds.)

[conr00] Maria Letizia Jaccheri and Reidar Conradi and Bard H. Dyrnes, *Software Process Technology and Software Organisations*, "European Workshop on Software Process Technology", p96-108,2000,

[Cang01] Joao W. Cangussu, *Modeling and Controlling the Software Test Process*, IEEE, International Conference on Software Engineering 2001, p 787-788

[Ghez98] Gianpaolo Cugola and Carlo Ghezzi, *Software processes: a retrospective and a path to the future* journal Software Process: Improvement and Practice, vol4 no3 p.1001-123 1998

[ISO9126] Draft international standard ISO/IEC 9126, *Information technology - Software system evaluation - Quality characteristics and guidelines for their use*, ISO International Organization for Standardization, International Electrotechnical Commission, 1996

[HLH92a] H.L.Hausen, A Software Assessment an Certification Advisor, in: J.P. Agrawal, V. Kumar, V. Wallentine (eds.), ACM, CSC'92 20th Annual Computer Science Conference, Kansas City, March 3-5, 1992

[McCa77] McCall, J.A.; Richards, P.K.; Walters, G.F. *Concepts and Definitions of Software Quality* Factors in Software Quality, Vol. 1, Springfield, Va.: NTIS, November 1977.

[Raea95] A. Rae, P. Robert, H.L. Hausen *Software Evaluation for Certification* McGraw Hill, London 1995

[RoMa89] Rombach, H. Dieter; Mark, Leo *Software Process & Product Specifications: A Basis for Generating Customized Software Engineering Information Bases*, Proceedings of the 22nd Annual Hawaii International Conference on System Sciences, Vol. II - Software Track, pp. 165-174, IEEE, 1989.